A GPS L1 and Cellular 4G LTE Vector Tracking Software-Defined Receiver

by

Samuel C. Morgan

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 9, 2023

Keywords: GPS, LTE, Software-Defined Receiver, Vector Tracking

Approved by

Scott Martin, Chair, Assistant Professor of Mechanical Engineering
Jitendra Tugnait, James B Davis Professor of Electrical and Computer Engineering
David Bevly, Bill and Lana McNair Distinguished Professor of Mechanical Engineering

Abstract

In this thesis, a traditional GPS L1 vector tracking receiver is augmented with cellular 4G Long-Term Evolution (LTE) channels. The hybrid receiver is capable of acquiring and tracking GPS L1 and LTE signals simultaneously in an Extended Kalman Filter (EKF) based vector tracking loop. The receiver tracks the Secondary Synchronization Signal (SSS) or the Cell-specific Reference Signal (CRS) to track the LTE signals.

First, the open-loop pseudorange and pseudorange rate errors variances are provided as a function of the carrier-to-noise ratio. A method to calculate the carrier-to-noise ratio of the LTE signal using correlator outputs is provided. The code phase and carrier frequency tracking accuracy of the receiver is derived analytically using the Discrete Algebraic Riccati Equation (DARE) and validated by Monte Carlo simulations. Next, the effects of errors, such as eNodeB localization error and multipath are evaluated in Monte Carlo simulation. Then, the robustness of the combined vector tracking loop to GPS outages is evaluated in Monte Carlo simulations. Finally, the positioning accuracy of the hybrid receiver is shown experimentally in two environments.

It is shown by DARE that the signal tracking performance of the combined receiver can improve significantly over GPS-only vector tracking receivers. The CRS can provide a larger performance increase than the SSS, but it is more susceptible to additional errors. It is shown that when other errors (e.g., multipath) dominate the LTE signals the overall position solution and, the signal tracking can be degraded significantly as well. The Monte Carlo simulations also show that the inclusion of LTE signals can improve the probability of tracking GPS signals during an outage period by up to 60%. However, if multipath is severe enough, tracking the SSS can decrease the probability of tracking the GPS signals.

In experiments, a simple way of mitigating errors due to eNodeB localization and multipath is implemented. Positioning results improve by up to 34% in static tests and up to 29% in dynamic tests over standalone GPS vector tracking. It is found that tracking the LTE signal with the CRS always improved the positioning solution, whereas tracking the SSS improved the solution in only 2 out of 3 tests when compared with GPS vector tracking.

Acknowledgments

My time in the GAVLAB has been a great learning experience. I owe a debt of gratitude to Drs. Dave Bevly and Scott Martin for facilitating that experience. I met many great people in the GAVLAB. In particular, trivia nights with the Dead Reckoners (you know who you are) are fond memories. Playing tennis/ping pong/pickleball with Connor Jones, Tyler Long, Noah Miller, Tanner Koza, Austin Smith, and Tahn Thawainin served as a great distraction from poring over this thesis. I especially owe Tanner thanks for his direct involvement with this work; I think my simulations would still be running otherwise. I appreciate Noah for reading over this thesis. I am also thankful that Tahn decided to come to Auburn and for being a great roommate. Finally, I'd like to thank my family, without whom none of this would have been possible.

Table of Contents

Abstract ii
Acknowledgments iv
List of Figures viii
List of Tables xv
List of Abbreviations xvi
1 Introduction 1
  1.1 Background and Motivation 1
  1.2 Prior Art 2
    1.2.1 Vector Tracking 2
    1.2.2 Signals of Opportunity 3
    1.2.3 Cellular Signals 4
  1.3 Research Contributions 5
  1.4 Thesis Outline 6
2 GPS and LTE Signal Stucture 7
  2.1 The GPS L1 C/A Signal 7
    2.1.1 Spreading Code 10
    2.1.2 Data Message 11
  2.2 The LTE Signal 13
    2.2.1 Frame Structure 13
    2.2.2 Ranging Signals 17
    2.2.3 Received Signal Model 19
3 GPS and LTE Software-Defined Receivers 22
  3.1 GPS Software-Defined Receiver 22

3.1.1 Signal Acquisition 24
3.1.2 Tracking Loops 26
3.2 LTE Software Defined Receivers 31
3.2.1 Signal Acquisition 31
3.2.2 System Information Decoding 37
3.2.3 Acquisition Refinement 39
3.2.4 LTE Signal Tracking 46
3.3 RF Positioning 51
3.3.1 Measurement Generation 52
3.3.2 Navigation Filter Design 55
4 Vector Tracking Loops 57
4.1 Vector Delay/Frequency Lock Loop 58
4.1.1 Extended Kalman Filter 58
4.2 GPS and LTE VDFLL 63
4.2.1 Combined Architecture 63
4.2.2 Measurement Residual Calculation 65
5 Simulation and Experimental Results 71
5.1 Simulations 72
5.1.1 Simulation Setup 72
5.1.2 Simulation Methodology 76
5.1.3 Simulation Results 80
5.2 Experiments 103
5.2.1 Experimental Setup 103
5.2.2 Experimental Results 108
6 Conclusion and Future Work 115
6.1 Conclusion 115
6.2 Future Work 116

Bibliography 119

Appendices 126

A 4 eNodeB AWGN Simulation Results 127

B 4 eNodeB Multipath Simulation Results 132

List of Figures

2.1 An example of BPSK modulation. 9

2.2 Autocorrelation Function of Gold Codes 10

2.3 Effect of the Spreading Code on Carrier in GPS L1 11

2.4 Arrangement of the GPS Navigation Message [34] 12

2.5 Block Diagram of OFDMA Modulation 14

2.6 LTE FDD Frame Structure [35] 15

2.7 An Example of an LTE Frame with $N_r = 72$ [22] 16

2.8 An Example of an LTE Frame with $N_r = 72$ in the Frequency Domain 16

3.1 SDR Block Diagram 23

3.2 Block Diagram of the Parallel Code Search Acquisition Algorithm 25

3.3 Ambiguity Functions with and without a GPS Signal 26

3.4 Generalized Block Diagram of the PLL/DLL 27

3.5 Common GPS Receiver Correlator Structure 28

3.6 Diagram of a Typical GPS Tracking Loop 30

3.7 Autocorrelation (left) and Cross-Correlation (right) of the Time-Domain PSS . 33

3.8 Autocorrelation (left) and Cross-Correlation (right) of the Time-Domain SSS . . 35

3.9 LTE Acquisition Block Diagram 36

3.10 Normalized PSS and SSS Acquisition Results with Real LTE Signal Data 36

3.11 MIB Coding Steps [22] 37

3.12 Tail Biting Convolution Encoder used in MIB Encoding 38

3.13 MIB Channel Decoding Method [51] 39

3.14 Mapping of the CRS for a Normal Cyclic Prefix [35] 41

3.15 Estimated CIR of the Same Data at Different Points in Time 43

3.16 Block Diagram of the FLL-Assisted PLL 48

3.17 Block Diagram of the CRS Tracking Loop 50

4.1 The classical VDFLL block diagram. 59

4.2 Vector and scalar tracking algorithms, subjected to a 60-second partial outage, beginning at $t = 15s$. 62

4.3 The full GPS and LTE VDFLL architecture block diagram. 64

5.1 Simulation Environment 73

5.2 Static (left) and Dynamic (right) Skyplots. 74

5.3 Monte Carlo simulation block diagram. 78

5.4 4 GPS and 1 eNodeB static simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the SSS. The GPS tracking results are shown on the right. 82

5.5 Probability of tracking LTE (left) and GPS (right) for the 4 GPS and 1 eNodeB static simulations. The LTE signals are tracked using the SSS. 83

5.6 5 GPS and 1 eNodeB dynamic simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the SSS. The GPS tracking results are shown on the right. 84

5.7 Probability of tracking LTE (left) and GPS (right) for the 5 GPS and 1 eNodeB dynamic simulations. The LTE signals are tracked using the SSS. 85

5.8 4 GPS and 1 eNodeB static simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the CRS. The GPS tracking results are shown on the right. 86

5.9 Probability of tracking LTE (left) and GPS (right) for the 4 GPS and 1 eNodeB static simulations. The LTE signals are tracked using the CRS. 86

5.10 5 GPS and 1 eNodeB dynamic simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the CRS. The GPS tracking results are shown on the right. 87

5.11 Probability of tracking LTE (left) and GPS (right) for the 5 GPS and 1 eNodeB dynamic simulations. The LTE signals are tracked using the CRS. 87

5.12 4 GPS and 4 eNodeB static simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the SSS. The GPS tracking results are shown on the right. 89

5.13 Probability of tracking LTE (left) and GPS (right) for the 4 GPS and 4 eNodeB static simulations. The LTE signals are tracked using the SSS. 89

5.14 5 GPS and 4 eNodeB dynamic simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the SSS. The GPS tracking results are shown on the right. 90

5.15 Probability of tracking LTE (left) and GPS (right) for the 5 GPS and 4 eNodeB dynamic simulations. The LTE signals are tracked using the SSS. 91

5.16 4 GPS and 4 eNodeB static simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the CRS. The GPS tracking results are shown on the right. 92

5.17 Probability of tracking LTE (left) and GPS (right) for the 4 GPS and 4 eNodeB static simulations. The LTE signals are tracked using the CRS. 92

5.18 5 GPS and 4 eNodeB dynamic simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the CRS. The GPS tracking results are shown on the right. 94

5.19 Probability of tracking LTE (left) and GPS (right) for the 5 GPS and 4 eNodeB dynamic simulations. The LTE signals are tracked using the CRS. 94

5.20 The code phase tracking performance of the 4 GPS and 1 eNodeB static simulations (left) and 5 GPS and 1 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS 96

5.21 The probability of tracking the LTE signal during the 4 GPS and 1 eNodeB static simulations (left) and 5 GPS and 1 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS. 96

5.22 The code phase tracking performance of the 4 GPS and 1 eNodeB static simulations (left) and 5 GPS and 1 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS 97

5.23 The probability of tracking the LTE signal during the 4 GPS and 1 eNodeB static simulations (left) and 5 GPS and 1 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS. 98

5.24 The code phase tracking performance of the 4 GPS and 4 eNodeB static simulations (left) and 5 GPS and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS 98

5.25 The probability of tracking the LTE signal during the 4 GPS and 4 eNodeB static simulations (left) and 5 GPS and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS. 99

5.26 The code phase tracking performance of the 4 GPS and 4 eNodeB static simulations (left) and 5 GPS and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS 100

5.27 The probability of tracking the LTE signal during the 4 GPS and 4 eNodeB static simulations (left) and 5 GPS and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS. 100

5.28 The probability of tracking the GPS signals throughout the outage simulations. On the left, the SSS is tracked. On the right, the CRS is tracked. 102

5.29 The equipment setup in the data collection vehicle. The GPS/LTE antenna is connected to two channels on the USRP X310. The GPS output is split and is fed to an onboard GPSDO in the USRP X310. The Pinwheel antenna is connected to the eTALIN for truth. 104

5.30 The ground truth static and dynamic trajectories for the experimental data recorded in Auburn, AL. 105

5.31 The ground truth trajectory for the experimental data recorded in Atlanta, GA. 105

5.32 The skyplot of the GPS satellites that are tracked in the experimental data recorded in Atlanta, GA. 106

5.33 Positioning performance of the GPS-only VDFLL and combined VDFLLs using the static data recorded in Auburn, AL. 109

5.34 The position errors in the east, north, and up axes for the static data collected in Auburn, AL. 110

5.35 Positioning performance of the GPS-only VDFLL and the combined VDFLLs using the dynamic data recorded in Auburn, AL. 111

5.36 The position errors in the east, north, and up axes for the dynamic data collected in Auburn, AL. 112

5.37 Positioning performance of the GPS-only VDFLL and the combined VDFLLs using the dynamic data recorded in Atlanta, GA. 113

5.38 The position errors in the east, north, and up axes for the dynamic data collected in Atlanta, AL. 114

A.1 4 eNodeB static simulation results. The LTE signals are tracked using the SSS. 127

A.2 Probability of tracking LTE in the 4 eNodeB static simulations. The LTE signals are tracked using the SSS. 128

A.3 4 eNodeB dynamic simulation results. The LTE signals are tracked using the SSS.128

A.4 Probability of tracking LTE in the 4 eNodeB static simulations. The LTE signals are tracked using the SSS. 129

A.5 4 eNodeB static simulation results. The LTE signals are tracked using the CRS. 129

A.6 Probability of tracking LTE in the 4 eNodeB static simulations. The LTE signals are tracked using the CRS. 130

A.7 4 eNodeB dynamic simulation results. The LTE signals are tracked using the CRS. 130

A.8 Probability of tracking LTE in the 4 eNodeB static simulations. The LTE signals are tracked using the CRS. 131

B.1 The code phase tracking performance of the 4 eNodeB static simulations (left) and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS 133

B.2 The probability of tracking the LTE signal during the 4 eNodeB static simulations (left) and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS. 133

B.3 The code phase tracking performance of the 4 eNodeB static simulations (left) and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS. 134

B.4 The probability of tracking the LTE signal during the 4 eNodeB static simulations (left) and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS. 134

List of Tables

2.1 Available LTE Bandwidths and Number of Subcarriers 14

4.1 Available bandwidths, sampling rates, and corresponding sample widths [69]. . . 66

5.1 Assigned center frequencies of each eNodeB. 74

5.2 The PDOPs of the possible configurations for each scenario. 74

5.3 eNodeB position estimation errors and multipath delay profiles [69] used for the simulations. 75

5.4 Multipath channel models [69] used in the presented simulations. 75

5.5 The PDOPs of the scenarios recorded in Atlanta, GA. 104

5.6 Static Experiment Statistics 110

5.7 Dynamic Experiment Statistics 111

5.8 Atlanta Statistics 114

List of Abbreviations

4G	Fourth Generation
AWGN	Additive White Gaussian Noise
BPSK	Binary Phase-Shift Keying
CDMA	Code-Division Multiple Access
CFR	Channel Frequency Response
CIR	Channel Impulse Response
CP	Cyclic Prefix
CRS	Cell-Specific Reference Signal
DARE	Discrete Algebraic Riccati Equation
DC	Direct Current
DFT	Discrete Fourier Transform
DLL	Delay Lock Loop
EKF	Extended Kalman Filter
FDD	Frequency-Division Duplex
FFT	Fast Fourier Transform
FLL	Frequency Lock Loop
GNSS	Global Navigation Satellite Systems

GPS	Global Positioning System
ISI	Inter-Symbol Interferences
KF	Kalman Filter
LOS	Line of Sight
LTE	Long-Term Evolution
Mcps	Megachips per Second
NCO	Numerically-Controlled Oscillator
PLL	Phase Lock Loop
PNT	Position, Navigation, and Timing
PRN	Pseudorandom Noise
PSS	Primary Synchronization Signal
RB	Resource Block
RE	Resource Element
RF	Radio Frequency
RG	Resource Grid
RMSE	Root Mean Squared Error
SDR	Software-Defined Receiver
SOP	Signal of Opportunity
SSS	Secondary Synchronization Signal
TDD	Time-Division Duplex

UE	User Equipment

Chapter 1

Introduction

1.1 Background and Motivation

Almost all modern Position, Navigation, and Timing (PNT) systems have relied on Global Navigation Satellite Systems (GNSS) for the past several decades. GNSS provide an absolute position fix. As opposed to a dead reckoned position, GNSS positions do not degrade over time. To produce a position fix, absolute timing must be known at the transmitter and estimated at the receiver. As such, GNSS are a heavily relied upon timing source for many critical applications.

These inherently weak signals are heavily attenuated indoors and in urban canyons. They are also susceptible to interference. As a result, they will not be able to meet the needs of emergent technologies such as autonomous vehicles, location-based services, and intelligent transportation systems [1]. Due to this inadequacy, many researchers fused GNSS receivers with inertial navigation systems (INS), lidars, cameras, or map matching algorithms [2], [3], and [4]. While others have used feedback from the navigation processor directly in a vector tracking loop [5], [6], and [7]. Although these methods have offered marked improvement over GNSS scalar tracking alone, they all assume that (1) numerous GNSS signals are present, and (2) one or more GNSS signal channels are of high quality.

The motivation for this thesis is to address the above limitations by enhancing a GNSS receiver's performance in degraded and limited signal environments. By improving receivers, PNT becomes more widely available, especially in urban canyons.

1.2 Prior Art

GNSS receivers are unable to produce measurements in difficult signal conditions because the receiver is unable to process the satellite transmissions and extract quality navigation observables. Since the late 1970s, significant research has been centered on improving the performance of GNSS receivers.

In the 1990s a new paradigm for contending with poor signal conditions arose in vector tracking. With this approach, high-quality channels can aid in the tracking of lower-quality channels. This approach still requires at least four satellites and at least one high-quality channel. Over the past decade, research has been focused on addressing the limitations of GNSS by exploiting ambient signals of opportunity (SOPs). The following section will discuss the above two areas of research in detail.

1.2.1 Vector Tracking

The method of extracting navigation observables from a received signal is called signal tracking. In signal tracking, feedback from the tracked signal is used to update local oscillators to match the parameters of the incoming signal. Commonly, GNSS signals are tracked using scalar tracking. In scalar tracking, feedback from each tracked channel is filtered and applied independently. Then, the observables are provided to the navigation processor, and a position, velocity, and timing (PVT) solution is obtained. Scalar tracking receivers are computationally efficient and can produce accurate PVT solutions given good signal conditions. However, in challenging signal environments, scalar feedback can become unusable and a poor representation of actual tracking errors. As a result, a measurement cannot be produced, and when signal conditions improve it is harder to reacquire the signal.

To combat these issues, research began on a tracking loop that would use feedback directly from the navigation processor. In 1996 James Spilker from Stanford University proposed the vector tracking loop which utilized a vector delay lock loop (VDLL) [8]. In this approach, a Kalman filter optimal estimator is used to predict the receiver's state and provide feedback to the tracking loop. As vector tracking is computationally burdensome, there was no rigorous performance analysis of vector tracking loops at that point.

In the early 2000s, with improvements in computing occurring rapidly, performance analysis began with Don Benson from the MITRE Corporation [9]. Soon thereafter, significant research was conducted by Lashley and Bevly [5], [10], [11], and [12]. Their research centered on comparisons between scalar and vector tracking loops. Their research determined that vector processing had a 2-6 dB improvement in tracking over traditional scalar tracking depending on the receiver's dynamics.

Despite all the advantages discussed above, vector tracking does have some disadvantages. For one, vector tracking is more complicated and computationally intensive than scalar tracking. As a result, vector tracking is not commonly used in commercial receivers. Vector tracking loops also suffer from noise sharing. That is, weaker channels will degrade the tracking of stronger channels. Similarly, if all GNSS channels are degraded, neither vector tracking or scalar tracking will be able to track the signals.

1.2.2 Signals of Opportunity

SOPs are ambient radio frequency (RF) signals that are not designed for navigation. They are freely available, and they are available in GNSS-challenged environments. Despite being designed without navigation in mind, they can be exploited for navigation purposes. Many SOPs are terrestrial or much closer to Earth than GNSS signals resulting in higher received power. Cellular, digital television, AM/FM, Wi-Fi, and low-earth orbit (LEO) satellite signals are examples of these SOPs [13, 14, 15, 16, 17, 18].

The literature discusses the observability and estimability of navigation with SOPs given *a priori* knowledge [19], receiver design, and localization techniques with various SOPs [20], [21], and [22]. Among the discussed SOPs, cellular signals are of note due to their numerous desirable characteristics. Cellular signals will be discussed in the following.

1.2.3 Cellular Signals

Cellular signals are RF signals broadcast by terrestrial beacons for high-speed communications. As such, they possess several desirable characteristics for navigation. In [23] Zak Kassas discusses these characteristics in detail. In [23], he discusses their abundance, geometric diversity, high received power, large bandwidth, accessibility, and resistance to jamming and spoofing as advantages over other SOPs and GNSS. Over the past decade, cellular communications standards have evolved to transmit large amounts of data quickly. In tandem, multiple receivers to track the various iterations of these signals have been developed in the literature.

First, the potential of cellular code division multiple access (CDMA), which is the transmission standard of the third generation (3G) of cellular signals, was evaluated comprehensively, and software-defined receivers were developed for these signals by Khalife and Kassas [21], [24]. In addition, they showed that these signals can achieve sub-meter accuracy when applied to an unmanned aerial vehicle (UAV) at constant altitude [16].

Next, the Fourth Generation (4G) of cellular signals, which transmit using the long-term evolution (LTE) standard, began to be evaluated. Hybrid GNSS and LTE localization methods were proposed and evaluated [25], [26], and [27]. Then, the first Software-Defined Receiver (SDR) was developed by del Peral-Rosado and Zanier towards hybrid positioning [28]. They used the classical Van de Beek estimator [29] for acquisition and tracked the signal using a traditional delay-lock loop (DLL) and phase-lock loop (PLL). As the idea of using SOPs as standalone positioning systems became more attractive, researchers began developing standalone LTE receivers. Marco Driusso from u-blox developed a standalone acquisition and tracking algorithm by combining the ESPRIT [30] algorithm with an Extended Kalman Filter (EKF) [31], [32]. This method, called EKAT, was able to achieve a Root-Mean-Square Error (RMSE) of 31 m on a dynamic vehicle in urban Switzerland using up to nine eNodeBs.

Despite the accuracy of the EKAT receiver, a simple but accurate receiver design was still desired. As such, Shamaei began developing receivers that are computationally similar to GPS receivers [22]. The first receiver tracked the Secondary Synchronization Signal (SSS), a repeated code that is broadcast reliably in each radio frame. This receiver obtained a 2-D RMSE of 50 m with code phase measurements. The next receiver tracked the Cell-Specific Reference Signal (CRS). This code is scattered in time and frequency and offered the best performance with a 2-D RMSE of 9 m using only code phase measurements.

In this thesis, only code phase and Doppler frequency measurements will be considered. Some successful receivers have been developed using carrier phase measurements in [22], but that is outside the scope of this thesis.

1.3 Research Contributions

This research seeks to improve existing GPS-only vector tracking loops by including ambient LTE signals. With this aim, several academic contributions have been made. These can be summarized as follows.

- A description of the combined vector tracking loop and its architecture is provided.
- A method of computing the open-loop pseudorange and pseudorange rate covariances of the LTE signal in terms of C/N_0 is developed. To the author's knowledge, these calculations have only been provided in terms of signal-to-noise ratio in the existing literature. In addition, they have only been provided in terms of symbol timing and carrier phase error instead of pseudorange and pseudorange rate error.
- A means of determining the C/N_0 of the LTE signal using correlator outputs is supplemented. Existing methods typically determine the signal-to-noise ratio of the OFDM signal and do not use correlator outputs.
- Thorough simulations are performed, assessing the best and worse case performance of the combined vector tracking loop.

- Experimental GPS and LTE signal data are collected. A first look at the performance of a prototype hybrid VDFLL using the collected data is given.

1.4 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 introduces the signal structures of the GPS L1 C/A and Cellular LTE signals. Chapter 3 presents the traditional software receivers used for processing these signals and RF positioning methods. Chapter 4 provides the VDFLL algorithm and some adaptations for incorporating LTE signals. Chapter 5 gives simulation methods and results that quantify the performance gains in the combined receiver. It also presents the real-world experiments used in this thesis and their results. Finally, chapter 6 discusses the conclusions of the work and future work.

Chapter 2

GPS and LTE Signal Stucture

GPS is one of many global satellite constellations that provide users with navigation and timing services. The GPS constellation broadcasts several signals on different center frequencies. In particular, the GPS L1 C/A signal (or legacy signal) is most commonly used by commercial receivers. It is also the only signal broadcast by all operational satellites. For a receiver to make use of the legacy signal for navigation, the signal structure must be known.

Cellular LTE signals are ubiquitous and are most commonly used for high-speed communications. LTE signals are broadcast from eNodeBs at several different center frequencies. To exploit these signals for navigation the low-level signal model must be known at the receiver or User Equipment (UE). As LTE signals are SOPs, the UE will be unable to communicate with the eNodeB and will exploit broadcast reference signals to extract navigation observables. The reference signals must be known to the UE before broadcast without a subscription to the network. If known, a received signal model can be developed and a navigation receiver can be designed. The signal structure of the LTE signal is provided by the 3GPP. It is replicated here for clarity.

This chapter discusses the GPS L1 C/A signal and the cellular LTE signals in detail. Section 2.1 describes the GPS L1 C/A signal. Section 2.2 presents the low-level model of LTE signals and their frame structure.

2.1 The GPS L1 C/A Signal

GPS is a constellation of orbiting satellites that broadcast ranging signals to the Earth's surface. Receivers process these signals to determine their position, velocity, and timing.

GPS was launched by the Department of Defense in 1973 and was made fully operational by 1995. GPS satellites transmit both civilian and military signals. Military signals are encrypted and only made available to the Department of Defense. Civilian signals are free to use and will be the focus of this work. Currently, the GPS constellation broadcasts four civilian signals: L1 C/A, L2C, L5, and L1C. L1 C/A is referred to as the *legacy* signal and is broadcast by all operational satellites. L2C, L5, and L1C are only broadcast by newer satellites. Most receivers use the L1 C/A signal for navigation and timing. The software receivers in this work employ the L1 C/A signal which is discussed further in this section.

The GPS L1 C/A signal consists of a carrier signal modulated with a Pseudorandom Noise (PRN) code and a navigation data message. The L1 C/A carrier is broadcast at a center frequency of 1575.42 MHz, the PRN code is broadcast at 1.023 MHz, and the navigation message is broadcast at 50 Hz. The signal is modulated by first combining the data message and the PRN code using the exclusive-OR (XOR) operation. Then, the result is multiplied by the carrier wave. This method of modulation is referred to as Binary Phase-Shift Keying (BPSK) . Figure 2.1 shows a visual representation of this form of modulation. The received signal model is expressed as

$$s(t) = \sqrt{2C}D(t)G(t)cos(\omega t + \theta), \tag{2.1}$$

where C is signal power, $D(t)$ is the data message, $G(t)$ is the spreading code, ω is the carrier frequency in radians per second, and θ is the carrier phase in radians. The carrier frequency of GPS L1 C/A is 1575.42 MHz, placing this signal in the L band of RF. In general, GPS receivers use the PRN code to estimate their position, the carrier frequency to estimate their velocity, and the navigation data message to determine information about the transmitting satellite's position, velocity, and timing.

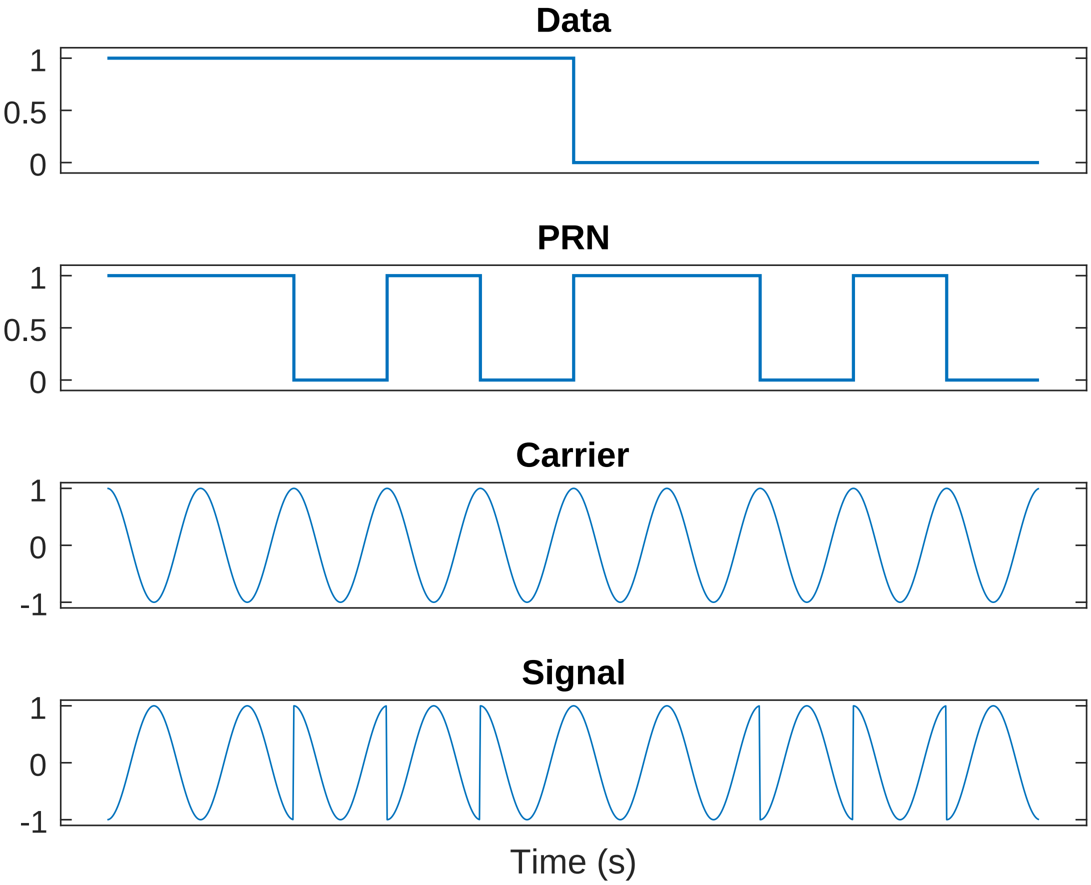

Figure 2.1: An example of BPSK modulation.

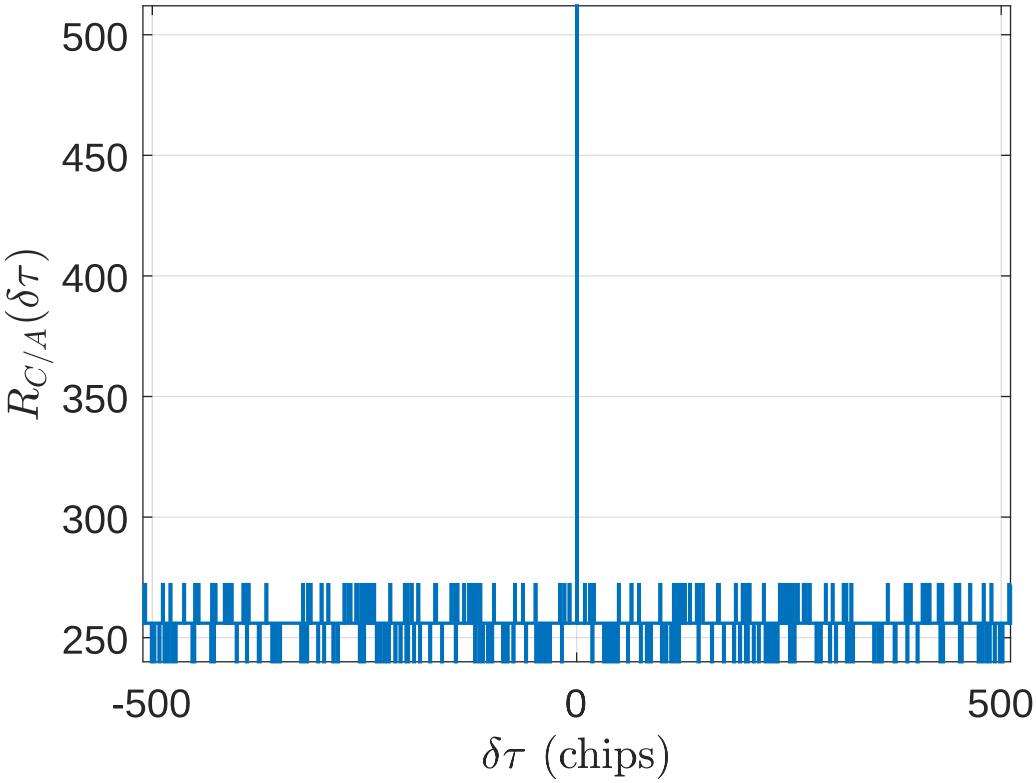

Figure 2.2: Autocorrelation Function of Gold Codes

2.1.1 Spreading Code

In the case of GPS L1 C/A, the spreading code, $G(t)$ is a Gold code [33] and is unique to each broadcasting satellite. Gold codes possess good correlation properties. Figure 2.2 shows the autocorrelation function of a Gold code. This autocorrelation function resembles that of Gaussian noise. That is, it correlates with itself at only one point in time and does not correlate with other sequences of the same type. For this reason, $G(t)$ is also referred to as a Pseudorandom Noise (PRN) code. This feature also makes the Code-Division Multiple Access (CDMA) properties of the GPS system possible.

The PRN code is generated at a chipping rate of 1.023 Megachips per Second (Mcps). The PRN code is also referred to as the spreading code. This is because of the effect of

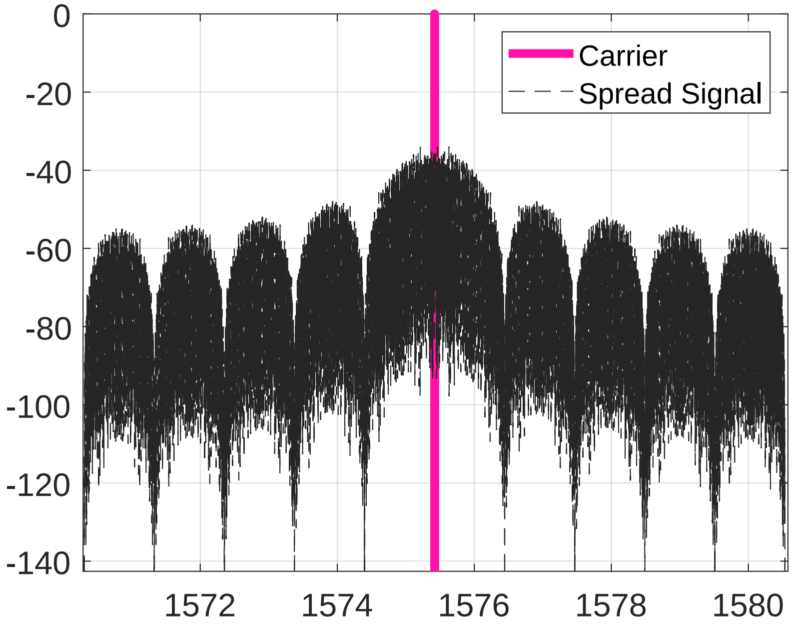

Figure 2.3: Effect of the Spreading Code on Carrier in GPS L1

applying this sort of modulation to a carrier sinusoid. The resultant signal is effectively "spread" in the frequency domain. Figure 2.3 shows the effect in the frequency domain compared with a sinusoidal carrier. The spreading code drops the carrier power by around 35 dB at the center frequency. This significant decrease in power, coupled with long transmission length ($\sim$ 20,000 km) causes the received signal to be well below the thermal noise floor.

2.1.2 Data Message

In addition to the spreading code, a data message is modulated onto the carrier in the same way. Unlike the spreading code, the bit rate of the data message is very low. The data message is sent at 50 Hz or one bit every 20 ms. That is, 20 full PRN sequences are sent during one data bit.

The data message contains information about the transmitting satellite. This information pertains to the satellite's orbit, health, GPS system time, and almanac data for the other satellites in the constellation. This message is arranged into thirty-bit words. 10 words

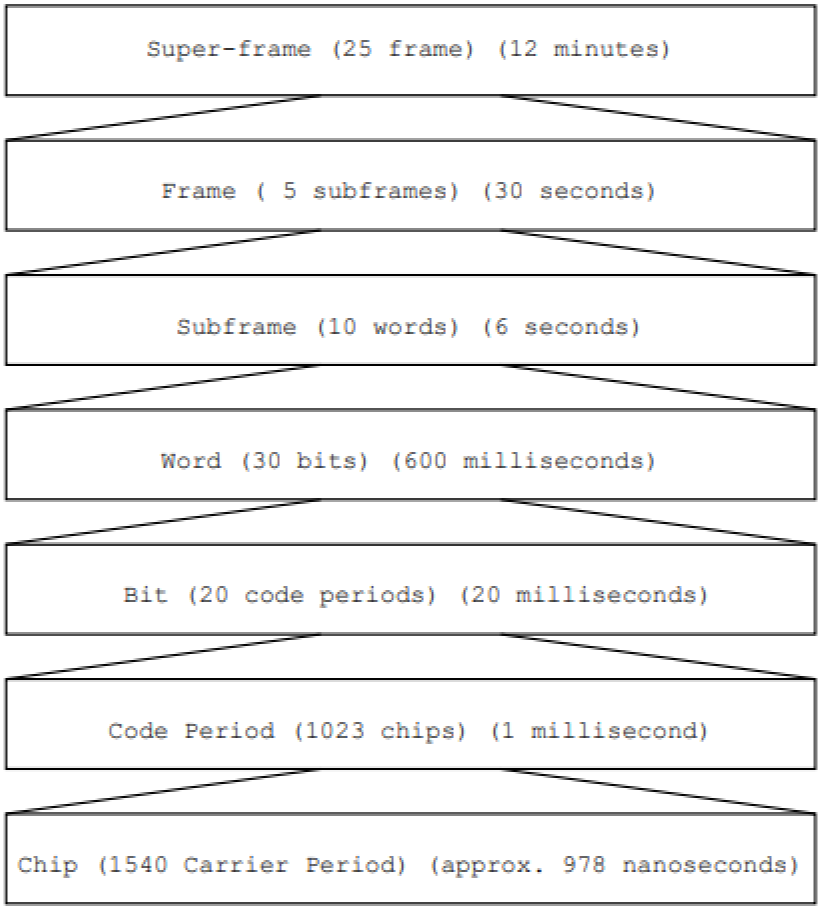

Figure 2.4: Arrangement of the GPS Navigation Message [34]

constitute a subframe and 5 subframes make a frame. Figure 2.4 shows how the navigation message is arranged. In each frame, the first 3 subframes contain clock and ephemeris information for the transmitting satellite. Subframes 4 and 5 alternately provide almanac data and ionospheric correction data. The satellite ephemeris data is information about the satellite's orbit. The ephemeris can be used to calculate the approximate position and velocity of the transmitting satellite. The almanac data is less precise ephemeris for other satellites in the constellation.

2.2 The LTE Signal

LTE is a standard for wireless broadband communications for mobile devices and data terminals. The standard was developed by the 3rd Generation Partnership Project (3GPP) and is specified in its Release 8 document series, with minor enhancements described in Release 9. The original idea for LTE was first proposed in 1998 to replace CDMA cellular signals. Studies officially commenced in 2005. The LTE standard was finalized in 2008, and the first publicly available LTE service was launched by TeliaSonera in Oslo and Stockholm. Afterward, it quickly became a mainstay in the rest of the world as well. Not long after, navigation with LTE signals became a popular research focus [25].

In order to exploit cellular LTE signals for navigation, the received signal model must be known by the User Equipment. In this thesis, navigation will be performed opportunistically. That is, the UE cannot communicate with the broadcasting eNodeB and must exploit the reliably transmitted reference signals. Knowing the structure of the transmitted LTE reference signals, the received signal can be modeled and a proper receiver structure to navigate with these signals can be designed.

2.2.1 Frame Structure

LTE systems transmit using orthogonal frequency division multiple access (OFDMA). In OFDMA, the transmitted symbols are mapped to orthogonal carriers called subcarriers. LTE uses a subcarrier spacing of $\Delta f = 15$ kHz. Each LTE base station or eNodeB broadcasts on N_r subcarriers. Each subcarrier contains a data symbol. Therefore, each serial data symbol must be mapped to N_r subcarriers. The mapping process depends on the LTE frame structure, where different data types are transmitted at different times and on different subcarriers. To reduce interference between neighboring frequency bands a guard band is allocated to LTE OFDMA signals, where no data is transmitted on the subcarriers on either side of the N_r data subcarriers. The guard bands must occupy, at most, 10% of the bandwidth of the transmitting signal. Hence, the total allocated bandwidth is not defined by N_r. Table

2.1 summarizes all available bandwidths in LTE and the number of data subcarriers. It should be noted that the Direct Current (DC) subcarrier is left intentionally empty.

Table 2.1: Available LTE Bandwidths and Number of Subcarriers

Total Bandwidth (MHz)	**Data Subcarriers ($\mathbf{N_r}$)**
1.4	72
3	180
5	300
10	600
15	900
20	1200

Once N_r is determined, the serial data symbols are mapped to each subcarrier in accordance with the LTE frame structure. Then, the DC subcarrier and guardbands are assigned zeros. An Inverse Discrete Fourier Transform (IDFT) is performed, resulting in a time-domain OFMDA symbol of duration $1/\Delta f$. The last L_{CP} elements of the OFDMA symbol are repeated at the beginning of the symbol to suppress Inter-Symbol Interference (ISI) due to multipath. L_{CP} is constant among all eNodeBs. These repeated elements are referred to as the Cyclic Prefix (CP) . Figure 2.5 shows the OFDMA modulation scheme for digital transmission, where $X_0, ..., X_{N_r-1}$ are the N_r modulated symbols in the frequency domain and $x_0, ..., x_{N_r-1}$ are the N_r time-domain symbols. The UE can then obtain the transmitted symbols by performing the steps below in reverse.

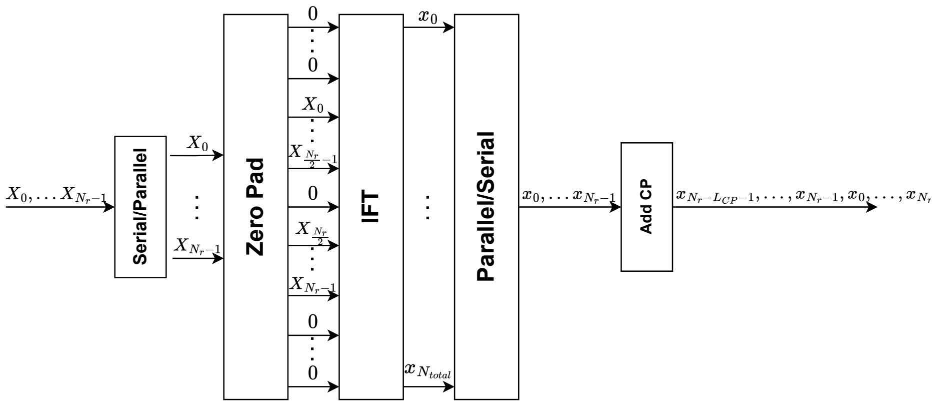

Figure 2.5: Block Diagram of OFDMA Modulation

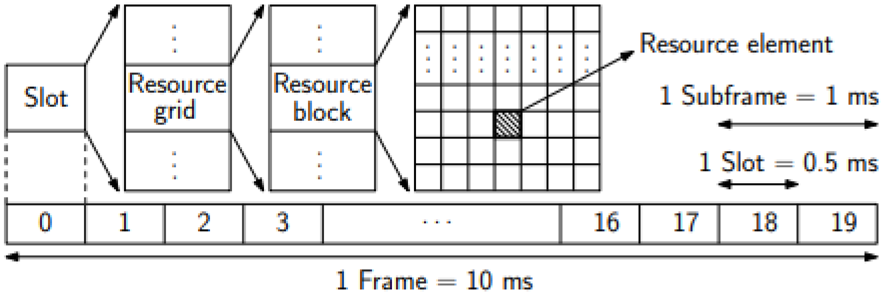

Figure 2.6: LTE FDD Frame Structure [35]

The resulting OFDMA symbols are grouped into frames of duration $T = 10$ ms. In LTE, the structure of the frame depends on the transmission scheme. There are two types of transmission schemes that are used in the LTE standard: Time-Division Duplex (TDD) , and Frequency-Division Duplex (FDD) . FDD is more commonly used [22] and will be the focus of this thesis.

Each frame consists of 10 subframes and 20 slots, with a duration of 1 ms and .5 ms, respectively. Each slot is made up of multiple Resource Grids (RGs). Each RG is composed of multiple Resource Blocks (RBs). The RB can be broken down to the smallest element in the radio frame, the Resource Element (RE). An RE can be defined two-dimensionally in frequency and time. The frequency and time indices of an RE are referred to as subcarrier and symbol, respectively. Figure 2.6 shows the structure of the LTE frame and Figure 2.7 shows an example of an LTE frame with $N_r = 72$. Figure 2.7 provides a top-down (time versus frequency) view of the LTE signal over time. Figure 2.8 provides a front-on view of the LTE frame in the frequency domain.

Since each RE is mapped to a specific subcarrier and symbol, the UE must first convert the received signal to the frequency domain. In particular, the specific symbol of interest must first be isolated. To do so, some knowledge of the frame timing must be known to the UE. Similar to GNSS, approximate frame timing can be determined by coarse acquisition. Figure 2.7 shows a number of reference signals, which will be discussed in the following subsection. The Secondary Synchronization Signal (SSS) and the Primary Synchronization

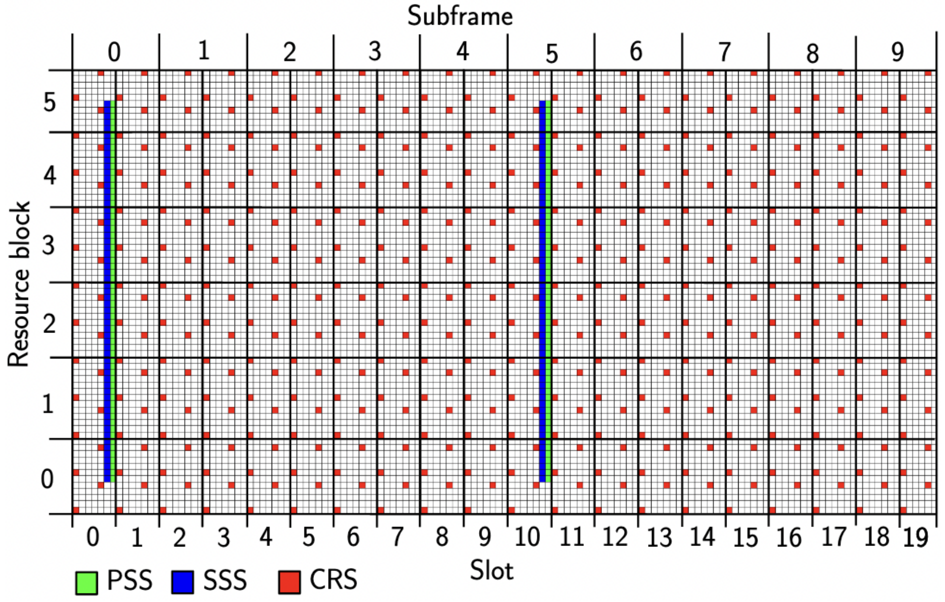

Figure 2.7: An Example of an LTE Frame with $N_r = 72$ [22]

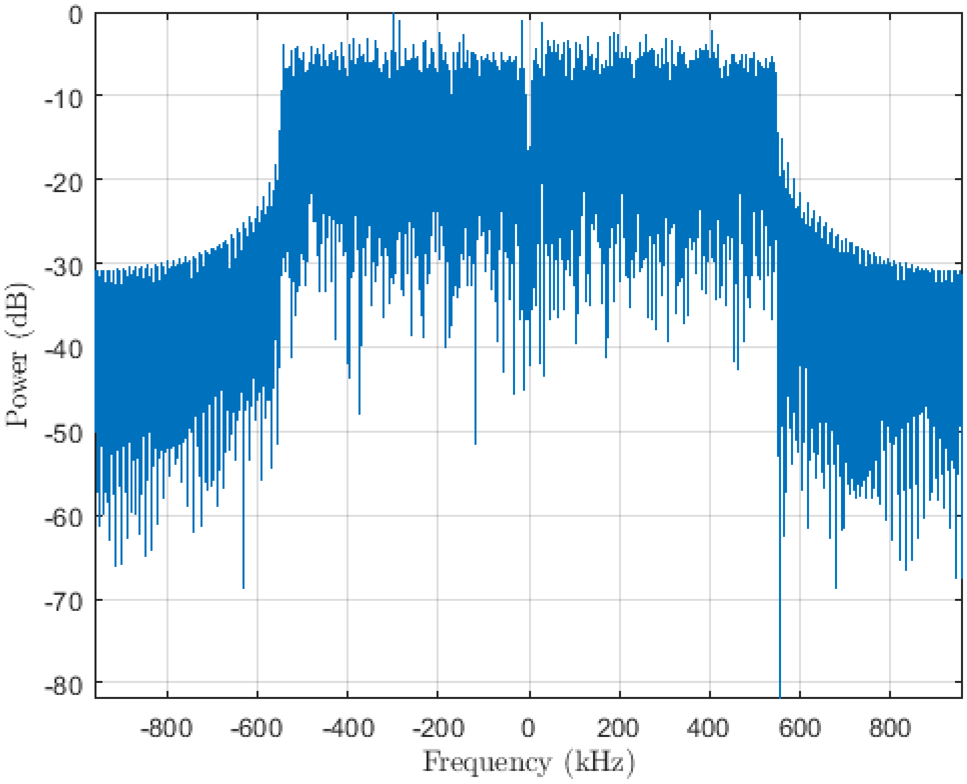

Figure 2.8: An Example of an LTE Frame with $N_r = 72$ in the Frequency Domain

Signal (PSS) are shown in blue and green, respectively. These signals can be correlated with locally generated replicas to determine frame timing. This same principle will be abused to perform signal tracking in the later sections of this thesis. Once frame timing is determined, the CP is removed and the UE can convert the symbol to the frequency domain. Then, the data on each of the N_r subcarriers can be demodulated and decoded.

2.2.2 Ranging Signals

Several sequences have been identified in the received LTE signals that can be used for positioning. In particular, two sequences have shown the most potential for navigation: the SSS and the Cell-Specific Reference Signal (CRS) [22]. Figure 2.7 shows the mapping of the SSS and a possible mapping of the CRS. Notably, both sequences are repeated reliably in every radio frame and the UE does not need to be a subscriber of the network to exploit them. Both sequences will be discussed in detail.

1. **SSS:** The SSS is an orthogonal length-62 sequence. It is transmitted in either slot 0 or 10 on the middle 62 subcarriers of the symbol preceding the PSS. The SSS is obtained by concatenating two maximal-length sequences scrambled by a third orthogonal sequence generated based on the PSS. As such, the PSS and the SSS are used together to determine frame timing and the cell ID of the eNodeB.

2. **CRS:** The CRS is transmitted for channel estimation purposes. As such, it is scattered in time and frequency. The sequence itself is determined by the cell ID, allocated symbol, slot, and transmission antenna port number. This ensures that different eNodeBs' CRS sequences are orthogonal to one another. In this thesis the transmitted CRS will be denoted as in [22] as $S_i(k)$ where $k = m\Delta_{CRS} + \kappa$, $m = 0, \cdots, M-1$, $M = [N_r/\Delta_{CRS}]$, $\Delta_{CRS} = 6$, and κ is a constant shift depending on the cell ID and the symbol number i.

In [22] and [36] receivers were developed for both the SSS and the CRS. In [31] and [32] receivers are developed for the CRS. Among the mentioned works, the CRS has been a much stronger candidate for navigation for a number of reasons. For one, the CRS occupies a much wider bandwidth than the SSS. As shown in Figure 2.7, the SSS only occupies the middle 62 subcarriers. The CRS will occupy as much bandwidth as is available. For example, an eNodeB that broadcasts on a 20 MHz bandwidth will broadcast a 20 MHz CRS. This makes the CRS much more robust to multipath compared with the 930 kHz bandwidth SSS. In addition, the CRS is broadcast more frequently than the SSS. The SSS is broadcast every 5 ms, whereas a CRS sequence is broadcast, approximately, every 0.25 milliseconds. This increased transmit period allows for higher resolution measurements and greater correlation power over longer integration periods. However, there are trade-offs. Namely, the computational complexity of tracking the CRS is much higher. The CRS is not found on contiguous subcarriers. As a result, no time-domain representation of the CRS is tractable. Therefore, the UE must be able to take many Discrete Fourier Transforms (DFTs) in real-time. Additionally, some information about the broadcasting eNodeB must be made available to the UE to make full advantage of the CRS. This information could be known to the UE if the environment has been studied previously, but this cannot be assumed. Hence, the UE must be able to decode data from the transmitting eNodeB to verify information, such as the transmitting antenna port number. Conversely, the SSS requires no DFTs and can be used completely opportunistically. Due to the trade-offs discussed here, the SDRs in this thesis will use either the SSS or the CRS to make measurements and the results will be compared.

2.2.3 Received Signal Model

In this section, the received signal models of the SSS and the CRS are presented.

SSS Received Signal Model

The SSS is transmitted twice per frame in slots zero and ten. However, the sequence is different depending on the slot. For both slots, we can denote the SSS in the frequency domain as $S_{SSS}(f)$. Since the SSS is fully contiguous in frequency, a one-to-one mapping from the frequency domain to the time domain is obtainable. Hence, the time domain representation $s(t)$ can be found as

$$s_{sss}(t) = \begin{cases} \text{IDFT}\{S_{\text{sss}}(f)\}, & \text{for } t \in (0, T_{\text{symb}}), \\ 0, & \text{for } t \in (T_{\text{symb}}, T), \end{cases} \tag{2.2}$$

It should be noted that the duration of each SSS replica is the duration of one LTE symbol, $T_{\text{symb}} = 1/\Delta f$. Then, assuming the transmitted signal is propagated in an Additive White Gaussian Noise (AWGN) channel, the received signal in the time domain can be written as [22]

$$\begin{aligned} r(t) = \sqrt{C}e^{j(2\pi f_D t+\Phi)}[s_{\text{sss}}(t - t_{\text{TOA}_k} - kT_f) + d(t - t_{\text{TOA}_k} - kT_f)] + n(t), \\ \text{for } kT_f \leq t \leq (k+1)T_f, \quad k = 0, 1, 2, ..., \end{aligned} \tag{2.3}$$

where k is the frame number, C is the received signal power including antenna gains and implementation loss, t_{TOA} is the time of arrival (TOA) of the SSS signal, Φ is the carrier phase, f_D is the total carrier frequency offset due to the Doppler shift, clock drift, and oscillators' mismatch, n(t) is AWGN with a constant power spectral density $N_0/2$ Watts/Hz, and $d(t)$ is some data transmitted by the eNodeB other than the SSS on the same symbol where

$$d(t) = 0 \quad \text{for } t \notin (t_{\mathrm{TOA}_k}, t_{\mathrm{TOA}_k} + T_{\mathrm{symb}}). \tag{2.4}$$

CRS Received Signal Model

The transmitted symbols that contain the CRS are defined by their subcarrier and OFDMA symbol index. A CRS symbol transmitted on the k-th subcarrier and on the i-th OFDMA symbol can be expressed as [22]

$$Y_i(k) = \begin{cases} S_i(k), & \text{if } k \in N_{\mathrm{CRS}}, \\ D_i(k), & \text{otherwise,} \end{cases} \tag{2.5}$$

where N_{CRS} denotes the set of subcarriers containing the CRS and $D_i(k)$ represents data symbols on other subcarriers.

It is important to note that each OFDMA symbol is assumed to travel through a multipath fading channel. This channel is assumed constant over the duration of a symbol. This channel can be described by its Channel Impulse Response (CIR) [37]. The CIR is expressed as

$$h_i(\tau) = \sum_{l=0}^{L-1} \alpha_{i,l} \delta(\tau - \tau_{i,l}), \tag{2.6}$$

where i is the symbol number, L is the number of multipath components, $\alpha_{i,l}$ is the complex attenuation of the l-th path, $\tau_{i,l}$ is the delay of the l-th path with respect to the first path, in Line of Sight (LOS) conditions $\alpha_{i,0} = 1$ and $\tau_{i,0} = 0$, and δ is the Dirac delta function. Finally, the received symbol after removing the CP and converting to the frequency domain, given perfect synchronization, is given by

$$R_i(k) = \sqrt{C} Y_i(k) H_i(k) + W_i(k), \qquad \text{for } k = 0, \cdots, N_c - 1, \tag{2.7}$$

where $R_i(k)$ is the received i-th symbol at the k-th subcarrier, C is the received signal power, $W_i(k) \sim \mathcal{CN}(0, \sigma^2)$, where $\mathcal{CN}(\mu, \beta)$ is the complex Gaussian distribution with mean μ and variance β, and

$$H_i(k) = \sum_{l=0}^{L-1} \alpha_{i,l} e^{-j2\pi\tau_{i,l}k/T_{\text{symb}}} \tag{2.8}$$

is the Channel Frequency Response (CFR).

It is unlikely that the received symbol timing will be perfectly synchronized with the transmitter. This can be due to clock drift, Doppler frequency, and/or carrier frequency offset. Then, assuming the time mismatch is less than the duration of the CP, the received signal at the i-th symbol can be rewritten as [38, 39]

$$R_i(k) = e^{j\pi e_f} e^{j2\pi(iN_t+L_{CP})e_f/N_c} e^{j2\pi e_\tau k/N_c} \sqrt{C} Y_i(k) H_i(k) + W_i(k), \qquad \text{for } k = 0, \cdots, N_c - 1, \tag{2.9}$$

where $N_t = N_c + L_{\text{CP}}$, $e_f = \frac{f_D}{\Delta f}$, and $e_\tau = \frac{t_{\hat{TOA}} - t_{TOA}}{T_s}$ is the symbol timing error normalized by the sampling interval $T_s = T_{\text{symb}}/N_c$. The first two exponentials in equation (2.9) model the effects of the carrier frequency offset and the third exponential models the effect of the symbol timing error. It is worth mentioning that the Doppler frequency for each subcarrier is slightly different due to their different frequencies. In this thesis, this difference is neglected and the Doppler frequency is defined with respect to the center frequency f_c as was done in [22].

Chapter 3

GPS and LTE Software-Defined Receivers

In order to use the signals presented in Chapter 2 for navigation, receivers must be designed to extract the necessary observables. For GPS, this field is mature and the basis of the SDR used in this thesis was originally published in [40]. For the reader's benefit, this SDR will be summarized in Section 3.1.

With respect to LTE signals, the field is nascent. LTE signals were not designed for navigation. To exploit these signals for navigation purposes, specialized receivers have been designed. The basis of the receivers used in this thesis was designed in [22]. These receivers will be discussed in detail in Section 3.2.

Finally, this chapter will conclude with a brief synopsis of RF positioning. Section 3.3 will cover the methods used to determine an initial position, velocity, and timing solution using the observables extracted by the receivers discussed in Sections 3.1 and 3.2.

3.1 GPS Software-Defined Receiver

In most real-world applications, hardware receivers are used to process GPS signals. This is because they operate in real-time and produce valuable PNT information on-the-fly. However, these receivers are often 'black box' devices, as the internal processes are not often known to the user. As a result, they are not easily modified. This makes hardware receivers parochial in the research context. To subvert this, some researchers have developed SDRs [40, 41]. SDRs can provide the same performance as hardware receivers, but they are easily customizable. SDRs are often designed for the Central Processing Unit (CPU) of a personal computer, allowing for widespread dissemination and development. Unfortunately, the processing of GPS signals involves several high-frequency tasks. CPUs are generally poor

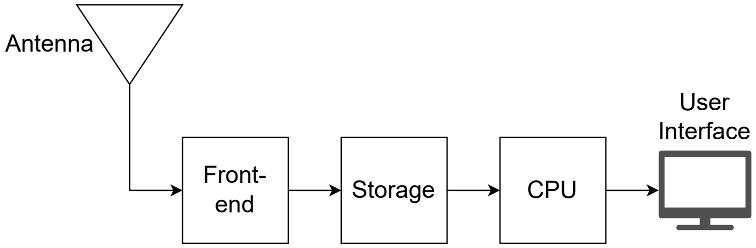

Figure 3.1: SDR Block Diagram

at computing high-frequency tasks, and as a result, most SDRs cannot operate in real-time. This has typically made SDRs not commercially viable since the information they provide will be out of date. Some researchers sought to overcome this by using reprogrammable hardware, such as Digital Signal Processors (DSPs), Field-Programmable Gate Arrays (FPGAs), or Graphics Processing Units (GPUs) [42, 43, 44]. These SDRs can operate in real time and offer relatively rapid reprogrammability. However, they are written in lower-level programming languages. Their results are often less accessible, and they are still more cumbersome to reprogram than CPU-based SDRs. For those reasons, this thesis will focus on CPU-based SDRs.

In general, an SDR has four primary components: the antenna, the RF front end, the processor, and the user interface. After the antenna, the analog signal is downconverted to an Intermediate Frequency (IF) and sampled by an Analog-to-Digital Converter (ADC). This entire process is handled by the RF front end. Then, the samples are sent to the processor that handles the receiver algorithms. In the case of a real-time SDR, this is done immediately. For most CPU-based SDRs the samples are first saved to storage, such as a hard disk or solid-state drive. Then, the samples are accessed at the CPU processing rate. The GPS SDRs used in this thesis operate in this fashion. The processor then uses software algorithms to acquire and track the GPS signals, while producing PNT solutions. Finally, the user interface displays the PNT information determined by the processor. Figure 3.1 shows a block diagram of an SDR. The SDR algorithms will be discussed in this section, beginning with signal acquisition.

3.1.1 Signal Acquisition

The first task of the SDR is to determine which satellites are transmitting and are in view. This task is often referred to as acquisition. In acquisition, three important parameters are discovered: the transmitting satellite ID, the code phase, and the Doppler frequency of the transmitting satellite. To do this, first consider the GPS received signal model in equation (2.1). This equation can be updated to be representative of the received signal, after downconversion, including satellite and receiver dynamics and clock offsets. It is rewritten as

$$s(t) = \sqrt{2C}D(t-\tau)G(t-\tau)cos(2\pi(f_D)t+\theta), \tag{3.1}$$

where C is the received signal power, τ, is the time delay or transit time between the satellite and the receiver due to distance and clock offsets, and f_D is the Doppler frequency caused by the relative motion and the clock offsets between the satellite and the receiver in Hertz. In Chapter 2, it is mentioned that each satellite broadcasts a different, uncorrelated Gold code, $G(t)$. Using this knowledge, the first step is to correctly identify $G(t)$. Since each Gold code is known by the receiver, the receiver can generate a replica of each Gold code and test its correlation with the received signal. Yet, Gold codes only correlate at one point in time. Hence, this correlation must be computed for each τ. This step effectively determines both the initial code phase and the broadcasting satellite. However, this correlation will only be high if $f_D < \frac{1}{2T_{\text{PDI}}}$, where T_{PDI} is the predetection integration time. This is not often the case as satellite dynamics can impart up to ±5 kHz of Doppler shift on static receivers on Earth, and T_{PDI} is usually one millisecond or higher. Therefore, τ and f_D must be found simultaneously. There are two main acquisition algorithms used for this. The serial search acquisition algorithm will check each possible code phase τ and each Doppler bin sequentially. This process is quite slow for CPU-based SDRs. The SDRs used in this thesis will make use of the parallel code search acquisition algorithm. In this algorithm, the Fast Fourier Transform (FFT) is used to search each τ simultaneously. Each Doppler bin is

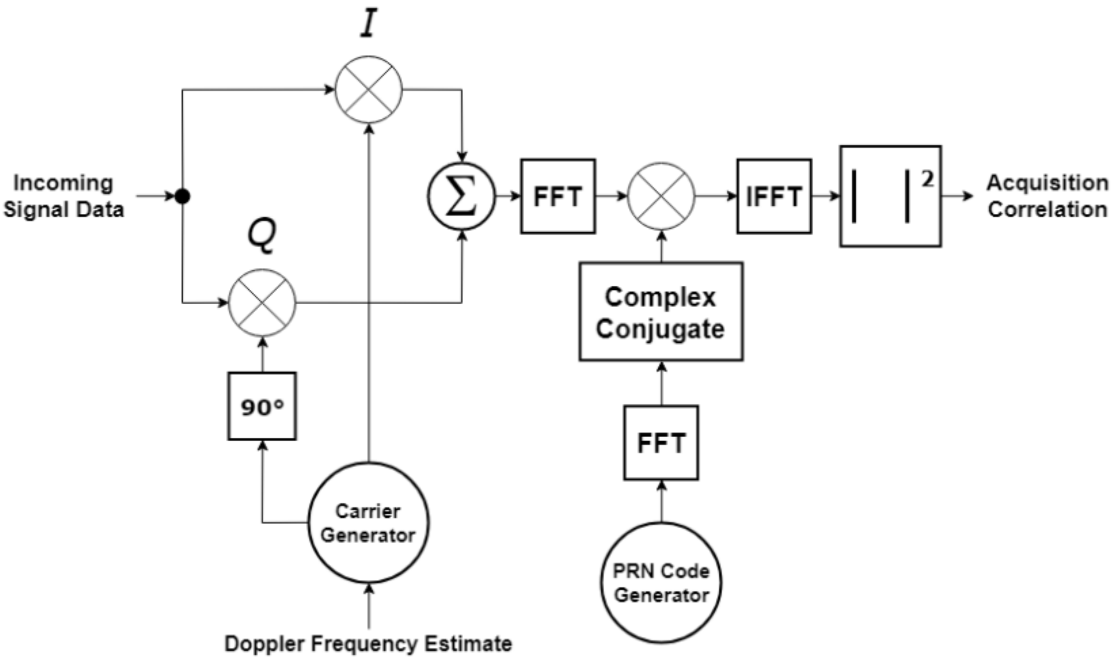

Figure 3.2: Block Diagram of the Parallel Code Search Acquisition Algorithm

still searched serially. The phrase Doppler bin is used here because searching each possible Doppler frequency is impossible. Besides, the residual Doppler frequency only must be lower than $\frac{1}{2T_{\text{PDI}}}$. Therefore, $\frac{1}{2T_{\text{PDI}}}$ is usually selected as the Doppler bin width. Figure 3.2 shows a block diagram of the parallel code search algorithm. Regardless of the algorithm used, the result is an ambiguity function. Figure 3.3 shows the resultant ambiguity function of a signal with a GPS signal present and one without a GPS signal present. The initial code phase and Doppler frequency are the x and y coordinates of the peak of the ambiguity function, respectively. These values are used to initialize the tracking loops that will be discussed in Subsection 3.1.2. More information on GPS acquisition algorithms can be found in [40] and [45].

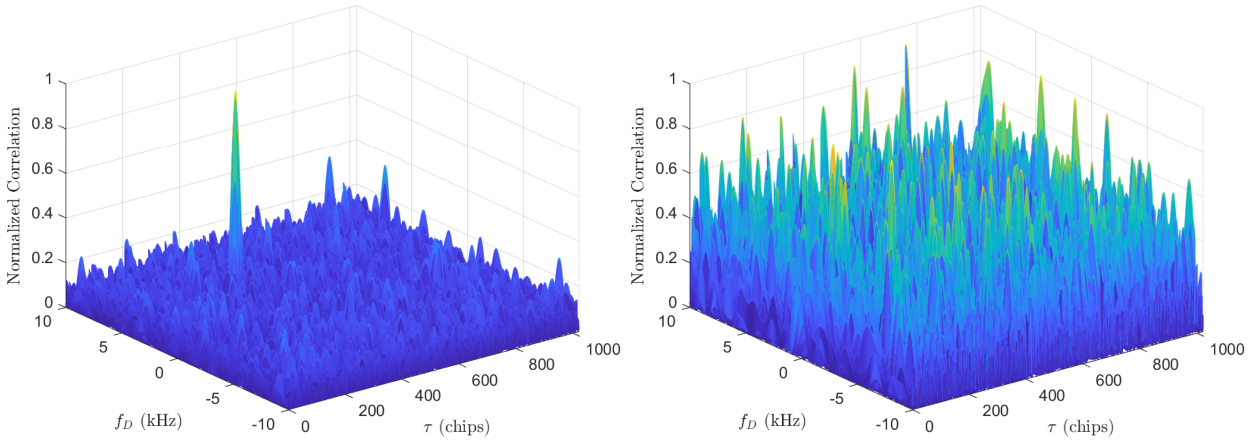

Figure 3.3: Ambiguity Functions with and without a GPS Signal

3.1.2 Tracking Loops

Once the initial conditions of the received signals are found, they must be updated according to the receiver and satellite spatiotemporal dynamics. Effectively, the SDR creates accurate local replicas of the received signal. In doing so, it can extract code and carrier phase and Doppler frequency measurements. These measurements can be used to calculate a PNT solution and will be discussed in detail in Section 3.3. These measurements could be extracted by repeating the acquisition steps discussed in Subsection 3.1.1, but to achieve the accuracy required for quality PNT solutions it would be computationally prohibitive. In other words, a more efficient way of generating these replicas is necessary. Tracking also is necessary to demodulate the data message $D(t)$ used to decode the ephemeris, which is necessary for navigation.

The typical method of signal tracking draws from control theory. That is, relatively simple controllers can be used to track small changes in code and carrier frequency. The receiver minimizes phase and frequency errors by adjusting the carrier frequency of the local replica. In general, the two controllers used for these functions are called Phase Lock Loops (PLLs) and Delay Lock Loops (DLLs), which track carrier and code phase, respectively. Both controllers use the same structure. Figure 3.4 shows a block diagram of the controller. The discriminator is a function that calculates the error of the local replica and the loop filter

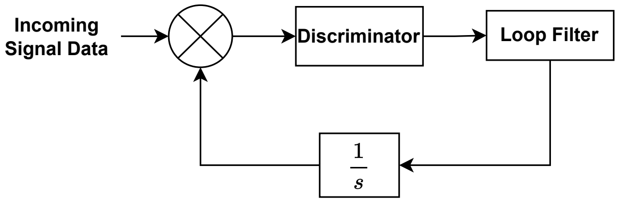

Figure 3.4: Generalized Block Diagram of the PLL/DLL

is a low-pass filter. For the PLL and the DLL, the error signal is the error in the carrier and code phase, respectively. The loop filter outputs a frequency correction, which is integrated using a Numerically-Controlled Oscillator (NCO) and used to generate a new local replica. The local replica is then correlated with the received signal and the process is repeated.

Figure 3.4, shows one of 6 total multiplications used in the tracking loop. The results of these correlations, or integrate and dumps, are often referred to as correlators. Figure 3.5 summarizes the six correlators commonly used in GPS receivers. The correlators are divided into in-phase and quadrature components by 90-degree carrier phase shift. Further, the correlators are separated into early, late, and prompt parts by delaying or advancing the code replica. The prompt code replica is the receiver's best estimate of the received code. The PLL uses the prompt correlators, and the DLL uses the early and late correlators.

The discriminator is different for both PLL and DLL. There are many viable functions for both carrier and code phase discriminators [45]. In general, choosing a discriminator function is often a trade between computational efficiency and accuracy. The most accurate ones are usually the most computationally inefficient. An example of carrier and code phase discriminators are shown in equations (3.2) and (3.3), respectively.

$$\delta\phi = \texttt{atan}(\mathrm{QP}/\mathrm{IP}) \tag{3.2}$$

$$\delta\tau = \frac{(\mathrm{IE}^2 + \mathrm{QE}^2) - (\mathrm{IL}^2 + \mathrm{QL}^2)}{2} \tag{3.3}$$

$\delta\phi$ is carrier phase error in radians, and $\delta\tau$ is code phase error in chips. Given that the correlator outputs themselves are inherently noisy, equations (3.2) and (3.3) will never report

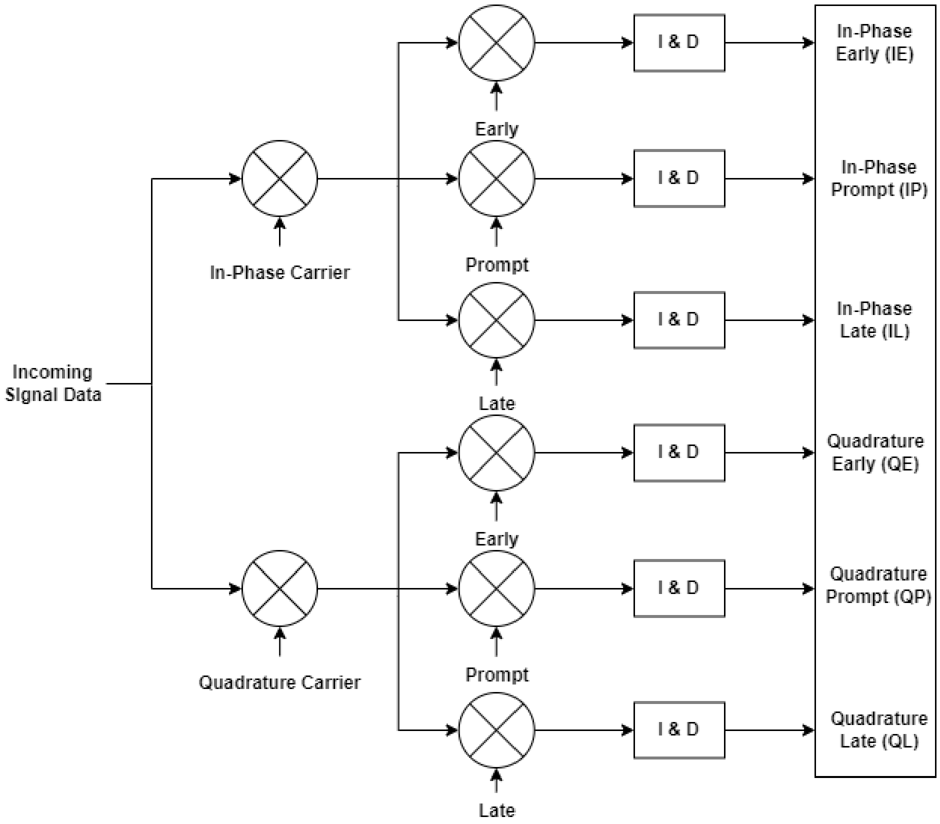

Figure 3.5: Common GPS Receiver Correlator Structure

the true error, but rather a noisy version of it. Equation (3.4) gives the models of the correlator outputs [46]

$$\begin{aligned}
\mathrm{IP} &= AR(\delta\tau)\mathrm{sinc}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}})\mathrm{cos}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}} + \delta\phi) + \eta_{\mathrm{IP}}, \\
\mathrm{QP} &= AR(\delta\tau)\mathrm{sinc}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}})\mathrm{sin}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}} + \delta\phi) + \eta_{\mathrm{QP}}, \\
\mathrm{IE} &= AR(\delta\tau + \Delta)\mathrm{sinc}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}})\mathrm{cos}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}} + \delta\phi) + \eta_{\mathrm{IE}}, \\
\mathrm{QE} &= AR(\delta\tau + \Delta)\mathrm{sinc}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}})\mathrm{sin}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}} + \delta\phi) + \eta_{\mathrm{QE}}, \\
\mathrm{IL} &= AR(\delta\tau - \Delta)\mathrm{sinc}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}})\mathrm{cos}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}} + \delta\phi) + \eta_{\mathrm{IL}}, \\
\mathrm{QL} &= AR(\delta\tau - \Delta)\mathrm{sinc}(\pi f_{\mathrm{err}} T_{\mathrm{PDI}})\mathrm{sin}(\pi f_{\mathrm{err}} T_{PDI} + \delta\phi) + \eta_{\mathrm{QL}}, \\
A &= \sqrt{2T_{\mathrm{PDI}}C}, \\
\eta &\sim N(0,1),
\end{aligned} \tag{3.4}$$

where f_{err} is the frequency error (Hz) between the received and locally generated carrier signals. The term C/N_0 is the carrier-power to noise-power spectral density ratio. The term Δ is the spacing between the correlators in chips. $R(\delta\tau)$ is the autocorrelation function of the received signal and local replica. In the case of GPS, the autocorrelation function is that of a Gold code. It is given as [45]

$$R(\delta\tau) = \begin{cases} 1 - |\delta\tau|, & \text{for } |\delta\tau| < 1, \\ 0, & \text{for } |\delta\tau| \geq 1. \end{cases} \tag{3.5}$$

Finally, Figures 3.4 and 3.5 are put together to form the full tracking loop. Figure 3.6 shows a block diagram of the tracking loop as computed by the CPU. After the stored signal is processed, the results of tracking can be used to calculate a PNT solution. This will be discussed in Section 3.3.

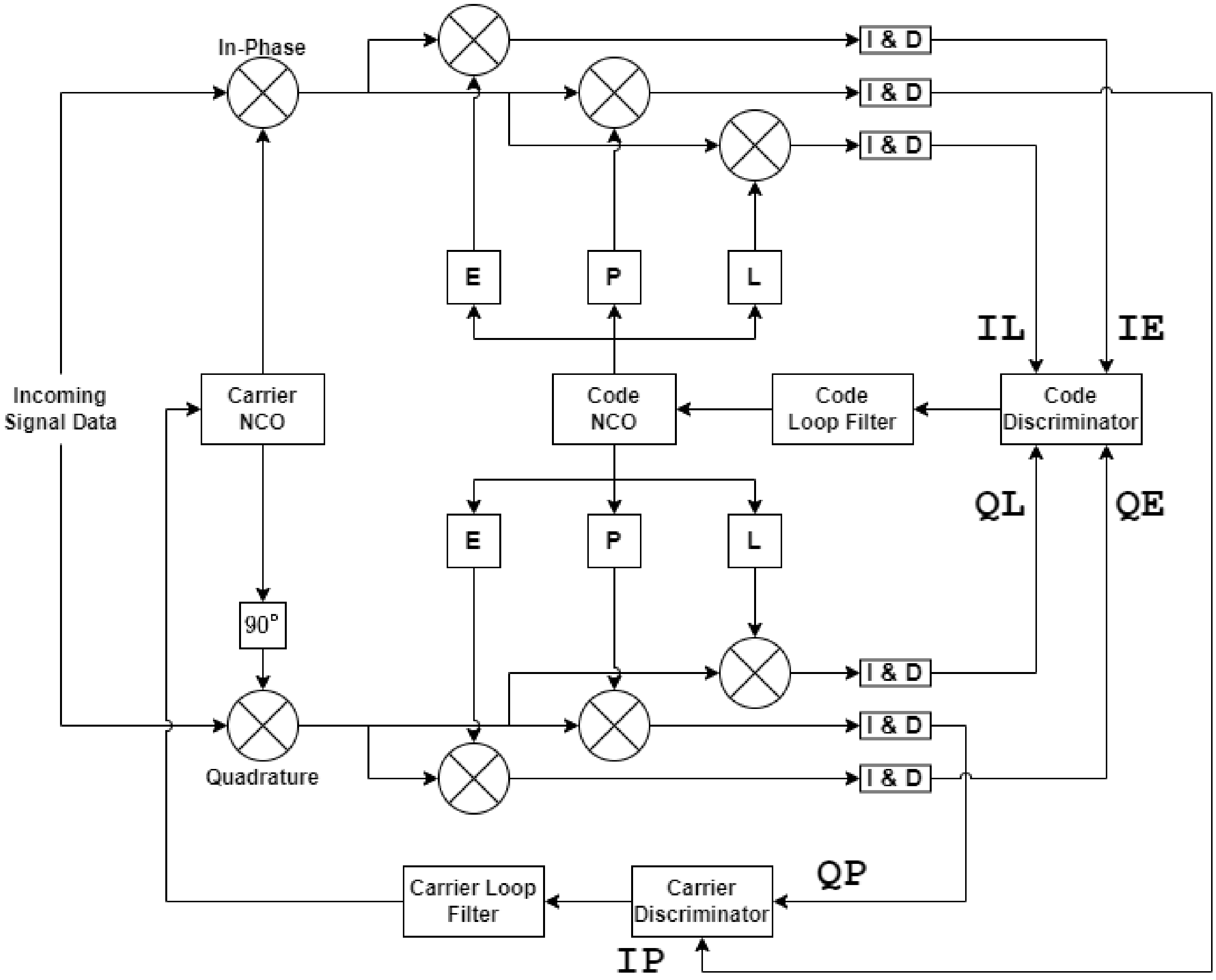

Figure 3.6: Diagram of a Typical GPS Tracking Loop

3.2 LTE Software Defined Receivers

Since navigation with SOPs is a recent development, very few hardware navigation receivers exist for LTE. In fact, most LTE navigation receivers are SDRs. The state-of-the-art code-phase LTE SDRs were developed in [22]. This section will discuss these SDRs in detail.

3.2.1 Signal Acquisition

The LTE SDR (UE) will begin with signal acquisition, like a GPS SDR. Each eNodeB is assigned a cell ID, $N_{\text{ID}}^{\text{cell}}$, which can be used to identify the transmitting eNodeB. The PSS and the SSS determine the cell ID. Since the PSS and SSS are contiguous in frequency, a time-domain representation of both can be generated. Three possible PSSs are available, and 168 possible SSSs are available. Since the SSS is dependent on the PSS, the PSS is acquired first. Once the time-domain representations of the PSS are generated, they are correlated with the received LTE signal. All PSSs with statistically significant peaks are determined to be present, representing $N_{\text{ID}}^{(2)}$. Once the transmitted PSSs are determined, the UE generates all possible SSS replicas for each acquired PSS. Then, the same correlation is performed using the SSS replicas. The SSSs with the highest correlation peaks are selected, representing $N_{\text{ID}}^{(1)}$. Finally, the cell ID can be calculated as

$$N_{\text{ID}}^{\text{cell}} = 3N_{\text{ID}}^{(1)} + N_{\text{ID}}^{(2)}, \tag{3.6}$$

where $N_{\text{ID}}^{(2)} \in \{0, 1, 2\}$ and $N_{\text{ID}}^{(1)} \in \{0, \ldots, 167\}$, depending on the PSS ($N_{\text{ID}}^{(2)}$) and SSS ($N_{\text{ID}}^{(1)}$).

PSS Generation

The PSS is a length-62 Zadoff-Chu sequence [47, 48]. It is distributed across the center 72 subcarriers (the DC subcarrier is empty) of the final symbol of slots 0 and 10. The five outermost subcarriers on either side are also left at 0. A Zadoff-Chu sequence is a complex sequence that is generated using the following

$$x_u(n) = \begin{cases} \exp(-j\frac{\pi u n(n+1)}{N}), & \text{for } n = 0, 1, \ldots, 30 \\ \exp(-j\frac{\pi u(n+1)(n+2)}{N}), & \text{for } n = 31, 32, \ldots, 62, \end{cases} \tag{3.7}$$

where N is the length of the sequence, which is specified as an odd positive integer (63 for LTE), and u is the root of the Zadoff-Chu sequence, specified as a positive integer. In the case of LTE, $u \in \{25, 29, 34\}$. This set is one-to-one with the set $N_{\mathrm{ID}}^{(2)} \in \{0, 1, 2\}$. For example, if the received LTE signal correlates with the Zadoff-Chu sequence generated with $u = 25$, then $N_{\mathrm{ID}}^{(2)} = 0$, and so on. The generated Zadoff-Chu sequence $x_u(n)$ is mapped to the middle 72 subcarriers and the result is $S_{\mathrm{pss}}(f)$. Finally, the PSS sequence is converted to the time-domain as

$$s_{\mathrm{pss}}(t) = \begin{cases} \mathrm{IDFT}\{S_{\mathrm{pss}}(f)\}, & \text{for } t \in (0, T_{\mathrm{symb}}), \\ 0, & \text{for } t \in (T_{\mathrm{symb}}, T). \end{cases} \tag{3.8}$$

Then, $s_{\mathrm{pss}}(t)$ has the correlation properties shown in Figure 3.7. Or, equivalently,

$$R_{\mathrm{pss}}(\delta\tau) \approx \mathrm{sinc}(\delta\tau), \tag{3.9}$$

where $\delta\tau$ is the code offset in chips.

SSS Generation

Once $N_{\mathrm{ID}}^{(2)}$ is found, the SSS can be generated. The SSS is obtained by using two binary sequences [35]. The sequences $s_0^{(m_0)}$ and $s_1^{(m_1)}$ are different cyclical shifts of an m-sequence, $\tilde{s}$. The indices m_0 and m_1 are determined by $N_{ID}^{(1)}$ according to

$$m_0 = m^{'} \mathrm{mod}\ 31, \tag{3.10}$$

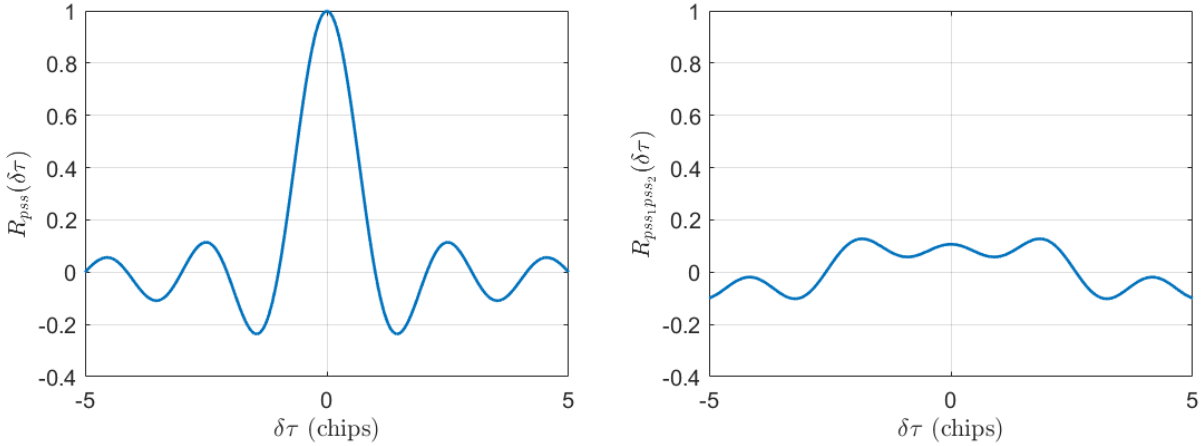

Figure 3.7: Autocorrelation (left) and Cross-Correlation (right) of the Time-Domain PSS

$$m_1 = (m_0 + \lfloor m^{'}/31 \rfloor + 1) \text{mod } 31, \tag{3.11}$$

where

$$m^{'} = N_{\text{ID}}^{(1)} + q(q+1)/2, \tag{3.12}$$

and

$$q = \left\lfloor \frac{N_{\text{ID}}^{(1)} + q^{'}(q^{'}+1)/2}{30} \right\rfloor, \quad q^{'} = \lfloor N_{\text{ID}}^{(1)}/30 \rfloor. \tag{3.13}$$

Then, the sequences $s_0^{(m_0)}$ and $s_1^{(m_1)}$ are generated according to

$$s_0^{(m_0)} = \tilde{s}((n + m_0) \text{mod } 31), \tag{3.14}$$

$$s_1^{(m_1)} = \tilde{s}((n + m_1) \text{mod } 31), \tag{3.15}$$

where $\tilde{s}(i) = 2 - x(i)$, $0 \leq i \leq 30$, is defined by

$$x(\bar{i} + 5) = (x(\bar{i} + 2) + x(\bar{i})) \text{mod } 2, \quad 0 \leq \bar{i} \leq 25 \tag{3.16}$$

with initial conditions $x(0) = 0$, $x(1) = 0$, $x(2) = 0$, $x(3) = 0$, $x(4) = 1$.

Then, these two sequences are scrambled by two sequences $c_0(n)$ and $c_1(n)$. These sequences depend on $N_{\text{ID}}^{(2)}$ and are defined by two different cyclic shifts of the m-sequence,

$\tilde{c}(n)$ according to

$$c_0(n) = \tilde{c}((n + N_{\text{ID}}^{(2)})\text{mod } 31), \tag{3.17}$$

$$c_1(n) = \tilde{c}((n + N_{\text{ID}}^{(2)} + 3)\text{mod } 31), \tag{3.18}$$

where $\tilde{c}(n)$ is generated in the same way as $\tilde{s}(n)$, except

$$x(\bar{i} + 5) = (x(\bar{i} + 3) + x(\bar{i})) \text{ mod } 2. \tag{3.19}$$

In addition to these sequences, the SSS symbols on odd subcarriers are scrambled by sequences $z_1^{(m_0)}(n)$ and $z_1^{(m_1)}(n)$. These sequences are defined by a cyclic shift of the m-sequence $\tilde{z}(n)$ according to

$$z_1^{(m_0)} = \tilde{z}((n + (m_0 \text{mod } 8)) \text{ mod } 31), \tag{3.20}$$

$$z_1^{(m_1)} = \tilde{z}((n + (m_1 \text{mod } 8)) \text{ mod } 31), \tag{3.21}$$

where $\tilde{z}(n)$ is obtained in the same way as $\tilde{s}(n)$ and $\tilde{c}(n)$, except

$$x(\bar{i} + 5) = (x(\bar{i} + 4) + x(\bar{i} + 2) + x(\bar{i} + 1)x(\bar{i})) \text{ mod } 2. \tag{3.22}$$

Finally, the sequence is generated according to the following

$$d(2n) = \begin{cases} s_0^{(m_0)}(n)c_0(n), & \text{if subframe 0,} \\ s_1^{(m_1)}(n)c_1(n), & \text{if subframe 5,} \end{cases} \tag{3.23}$$

and

$$d(2n+1) = \begin{cases} s_1^{(m_0)}(n)c_1(n)z_1^{(m_0)}(n), & \text{if subframe 0,} \\ s_0^{(m_1)}(n)c_1(n)z_1^{(m_1)}(n), & \text{if subframe 5.} \end{cases} \tag{3.24}$$

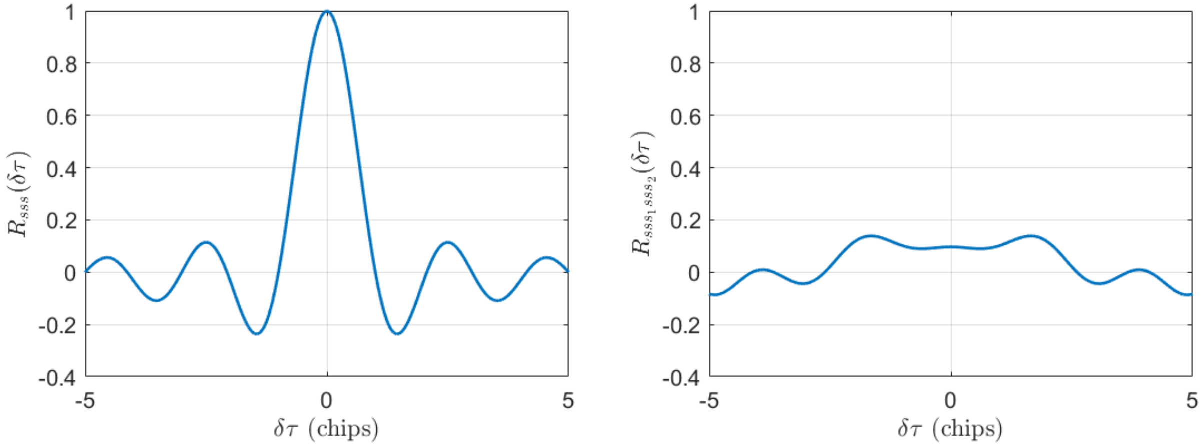

Figure 3.8: Autocorrelation (left) and Cross-Correlation (right) of the Time-Domain SSS

Then, $d(n)$ is mapped to the center 72 subcarriers and the result is $S_{\text{sss}}(f)$. The SSS sequence in the time domain is shown in equation (2.2). Figure 3.8 shows the correlation properties of $s_{\text{sss}}(t)$. Notably, the time-domain SSS and PSS have the same correlation properties. Equations (3.23) and (3.24) show that the SSS differs depending on the subframe. Therefore, the acquired SSS can be used to determine approximate frame timing.

Acquisition Scheme

In addition to the cell ID, acquisition is used to determine frame timing. Since the PSS is repeated twice in each frame it does not provide timing information. The SSS is repeated, but it differs depending on the subframe. Hence, it provides enough information to determine frame timing. Figure 3.9 shows a block diagram of the LTE acquisition scheme. First, the PSS is acquired and $N_{\text{ID}}^{(2)}$ is identified. Using $N_{\text{ID}}^{(2)}$, the SSS can be acquired. Finally, the approximate frame timing is extracted from the SSS acquisition and $N_{\text{ID}}^{\text{CELL}}$ is calculated. Figure 3.10 shows the results of the PSS and SSS acquisition blocks over a ten-millisecond radio frame on real LTE signal data. It is clear that the PSS is repeated and the SSS is not. Therefore, frame timing can be determined.

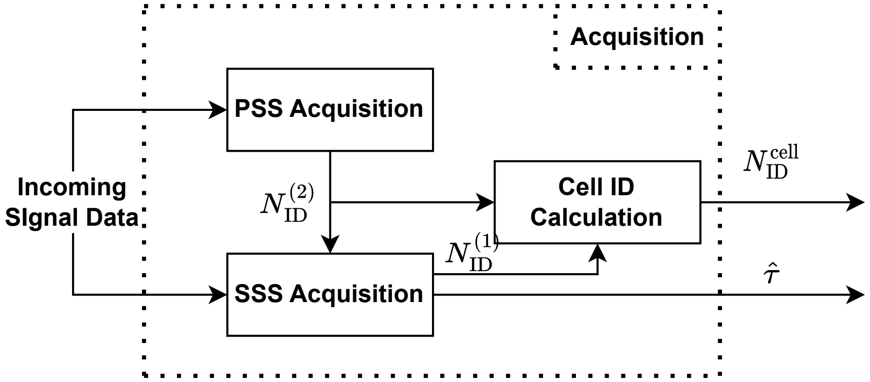

Figure 3.9: LTE Acquisition Block Diagram

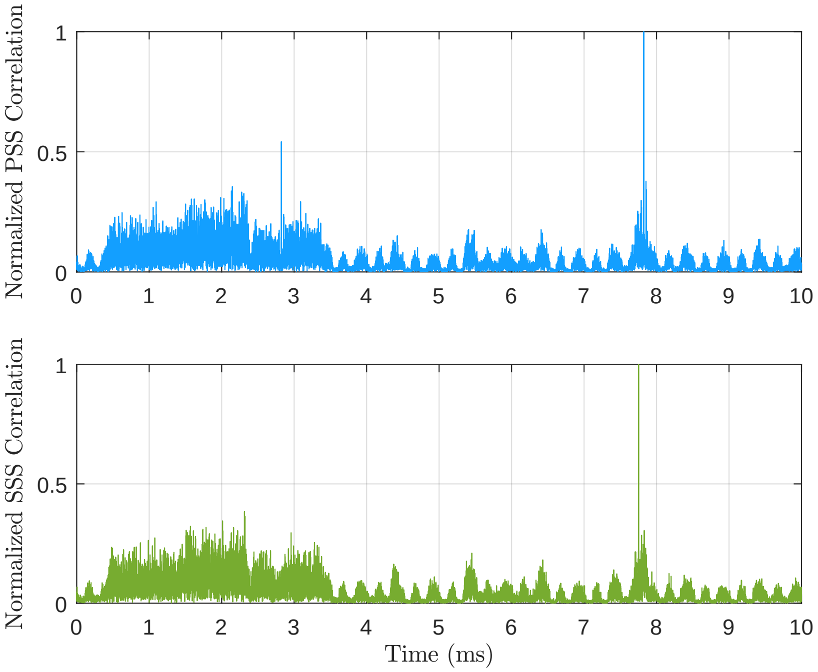

Figure 3.10: Normalized PSS and SSS Acquisition Results with Real LTE Signal Data

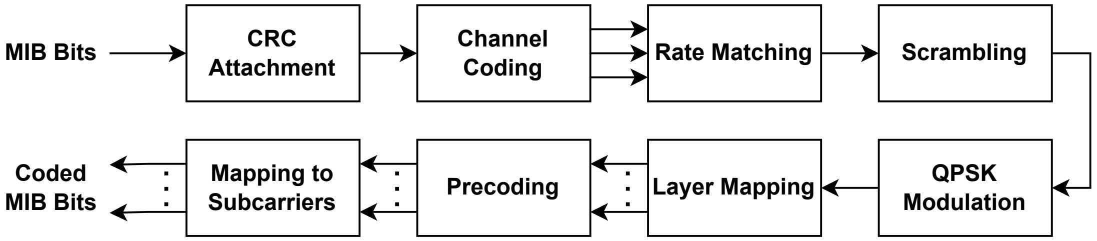

Figure 3.11: MIB Coding Steps [22]

3.2.2 System Information Decoding

After approximate frame timing is determined, the UE requires several parameters to exploit the LTE signal fully. It is worth noting that the SSS can be tracked at this stage since all relevant information is known. However, to use the high-fidelity CRS, some information about the transmitting eNodeBs must be extracted. In particular, the Master Information Block (MIB) must be decoded. It is transmitted on the Physical Broadcast Channel (PBCH).

MIB Decoding

To exploit the high-bandwidth CRS for navigation, the UE must know the transmission bandwidth and the number of transmitting antennas. In general, a mapping step could be performed first to ascertain the signal information prior to the navigation. In fact, it would be rather easy to determine the transmission bandwidth using a spectrogram, for example. For the experiments done in this thesis, the MIB information is determined *a priori*. Yet, it is assumed that most UEs will be in an unknown environment and must acquire this information on-the-fly. As such, the UE must be capable of decoding the MIB. The MIB is transmitted on the PBCH and consists of twenty-four bits of data. Three bits provide the downlink bandwidth, three bits provide the frame number, and eighteen bits provide other information. These bits are coded and sent on four consecutive symbols of the second slot of the frame. Figure 3.11 shows a block diagram of the MIB coding scheme [35, 50].

First, a Cyclic Redundancy Check (CRC) of length $L = 16$ is generated using the cyclic generator polynomial $g_{\text{CRC}}(D) = D^{16} + D^{12} + D^{5} + 1$. The number of transmitting antennas

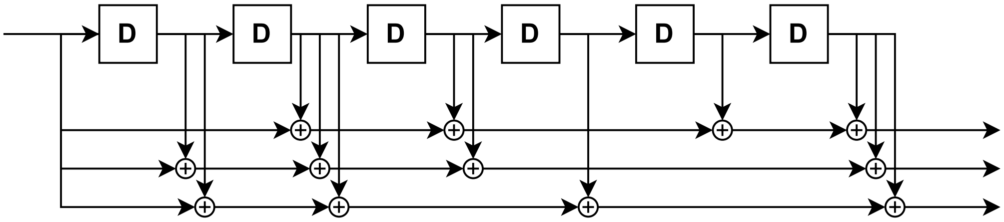

Figure 3.12: Tail Biting Convolution Encoder used in MIB Encoding

is not broadcast in the MIB. This information is provided in the CRC mask, a sequence used to scramble the CRC bits appended to the MIB. The CRC mask is either all zeros, all ones, or an alternating sequence of ones and zeros, corresponding to one, two, or four transmitting antennas, respectively. To determine the number of transmitting antennas, the UE performs a blind search over all possible CRC masks. That is, the UE must generate all possible CRCs given every CRC mask. Once the correct CRC is generated, the right number of transmitting antennas can be found.

In the second step, the MIB bits with CRC appended are coded with a convolutional encoder with a constraint length of seven and a coding rate of 1/3. Figure 3.12 shows the configuration of the convolutional encoder. The initial value of the encoder is set to the value of the last six information bits in the input stream.

The convolutionally encoded bits are then rate-matched. That is, the three output streams are interleaved. Then, the elements of the interleaved stream are repeated to obtain a 1920-bit array [50]. Finally, the output of the rate-matching step is scrambled with a pseudo-random sequence, which is initialized with the cell ID. The bits are then modulated using QPSK, resulting in 960 symbols. These symbols are mapped onto the predetermined subcarriers for MIB transmission [50].

To decode the received MIB, a traditional method of decoding a tail-biting convolutional code, like the one in [51] can be used. Figure 3.13 shows a block diagram of their method. The decoder is shown to be relatively simple and easily implementable on existing receiver designs.

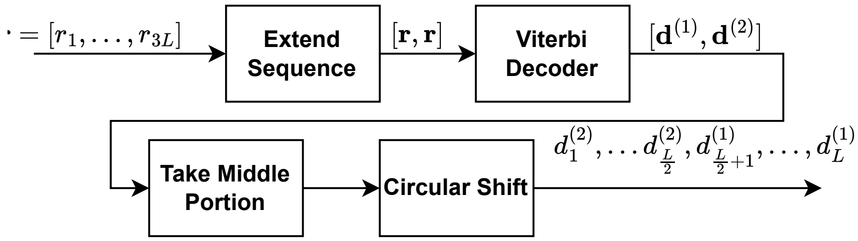

Figure 3.13: MIB Channel Decoding Method [51]

3.2.3 Acquisition Refinement

After the MIB is decoded, the subcarrier, symbol indices and symbols of the CRS can be determined. Since the CRS is known to the UE, the UE can exploit its wide bandwidth to provide a better estimate of the initial TOA and frequency offset.

CRS Generation

The reference signal $r_{l,n_s}(m)$ is defined by [35]

$$r_{l,n_s}(m) = \frac{1}{\sqrt{2}}(1 - 2c(2m)) + j\frac{1}{\sqrt{2}}(1 - 2c(2m+1)), \quad m = 0, 1, \ldots, 2N_{\text{RB}}^{\text{max,DL}} - 1 \quad (3.25)$$

where l is the OFDMA symbol within slot n_s. The pseudorandom sequence $c(i)$, discussed later, is initialized with $c_{\text{init}} = 2^{10}(7(n_s+1)+l+1) + 2N_{\text{ID}}^{\text{cell}} + N_{\text{CP}}$ for each OFDMA symbol where

$$N_{\text{CP}} = \begin{cases} 1, & \text{for normal CP,} \\ 0, & \text{for extended CP,} \end{cases} \quad (3.26)$$

and in this thesis, the normal CP will be assumed. The sequence $r_{l,n_s}(m)$ is then mapped to complex-valued symbols $a_{k,l}^{(p)}$ used as reference symbols for antenna port p in slot n_s according to

$$a_{k,l}^{(p)} = r_{l,n_s}(m'), \quad (3.27)$$

where [35]

$$
\begin{aligned}
k &= 6m + (v + v_{\text{shift}}) \bmod 6, \\
l &= \begin{cases} 0, N_{\text{symb}}^{\text{DL}} - 3, & \text{if } p \in \{0, 1\}, \\ 1, & \text{if } p \in \{2, 3\}, \end{cases} \\
m &= 0, 1, \ldots, 2N_{\text{RB}}^{\text{DL}} - 1, \\
m' &= m + N_{\text{RB}}^{\text{max,DL}} - N_{\text{RB}}^{\text{DL}},
\end{aligned}
$$

where $N_{\text{RB}}^{\text{max,DL}} = 110$ is the maximum number of downlink resource blocks, and, commonly, $N_{\text{symb}}^{\text{DL}} = 6$. $N_{\text{RB}}^{\text{DL}}$ is the number of downlink resource blocks in the received signal. Then, the variables v and v_{shift} define the position in the frequency domain for the different reference signals where v is given by [35]

$$
v = \begin{cases} 0, & \text{if } p = 0 \text{ and } l = 0, \\ 3, & \text{if } p = 0 \text{ and } l \neq 0, \\ 3, & \text{if } p = 1 \text{ and } l = 0, \\ 0, & \text{if } p = 1 \text{ and } l \neq 0, \\ 3(n_s \bmod 2), & \text{if } p = 2, \\ 3 + 3(n_s \bmod 2), & \text{if } p = 3, \end{cases} \tag{3.28}
$$

and the cell-specific frequency shift is given by $v_{\text{shift}} = N_{\text{ID}}^{\text{cell}} \bmod 6$. An example of how this allocation works is shown in Figure 3.14.

TOA Refinement

To further refine the TOA, the CRS can be used. First, consider the CIR discussed in Chapter 2. The CIR consists of complex attenuations and path delays. In LOS conditions,

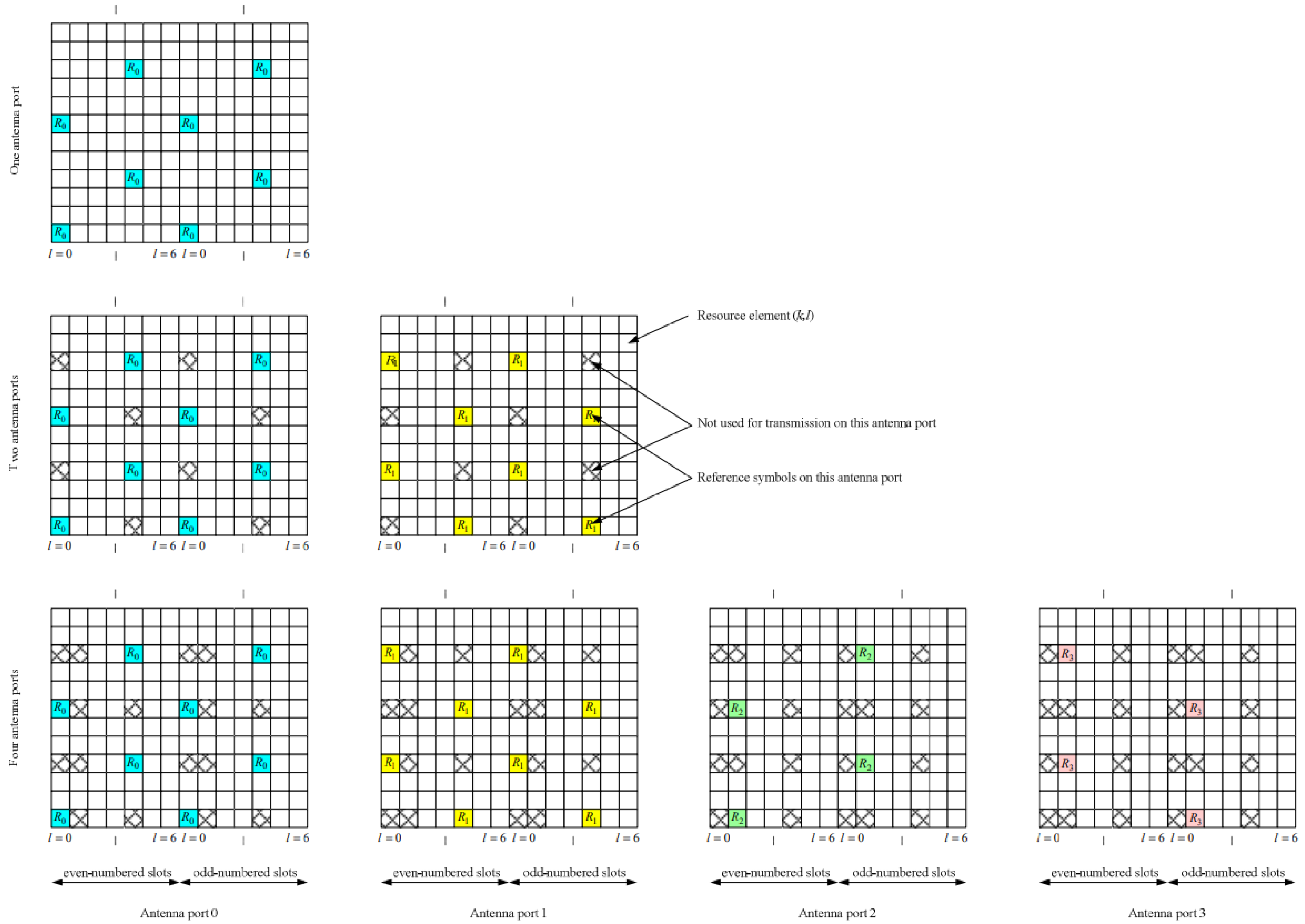

Figure 3.14: Mapping of the CRS for a Normal Cyclic Prefix [35]

the first path delay, $\tau_{i,0} = 0$ and the attenuation, $\alpha_{i,0} = 1$. However, if the acquired TOA is incorrect, $\tau_{i,0} \neq 0$, but the complex attenuation should remain the same. Therefore, by using the CIR the residual TOA error can be estimated and removed. Yet, the CIR is not directly observable to the UE. However, the CFR is observable with approximate frame timing and knowledge of the CRS. Equation (2.7) shows that the received CRS symbols are a linear combination of the CFR and the true CRS. Hence, the CFR can be estimated, in the least-squares sense, as

$$\hat{H}_i(k) = R_i(k)/Y_i(k) + W_i^{'}(k), \quad \text{for} \quad k \in N_{\text{CRS}} \tag{3.29}$$

where Ri(k) is the received symbol at subcarrier k and symbol i Yi(k), is the known transmitted symbol at subcarrier k and symbol i, and $W_i^{'}(k) = W_i(k)/Y_i(k)$, where $W_i(k)$ is complex Gaussian noise. Then, the CIR can be determined using the relationship

$$\hat{h}_i(\tau) = \text{IDFT}\{\hat{H}_i(k)\}, \tag{3.30}$$

where $h_i(\tau)$ is the CIR. Since $\alpha_{i,0} = 1$ the error in the time of arrival can be estimated as

$$\hat{e}_{TOA} = \arg\max_{\tau} \hat{h}_i(\tau) \tag{3.31}$$

where $\hat{e}_{\text{TOA}}$ is the error in the time of arrival in seconds. There are two main issues with this approach. For one, the resolution is limited to the frequency spacing of the CRS symbols. If the UE uses one OFDMA symbol, the spacing is $\Delta f_{\text{CRS}} = 90$ kHz. This results in the set of $\tau \in \{-5.56\mu\text{s}, \cdots, 5.56\mu\text{s}\}$. Which is further divided into the number of CRS symbols used in the CIR estimation. For example, for an LTE signal with a bandwidth of 10 MHz, there are 100 CRS symbols in each OFDMA symbol. As a result, there is a resolution of approximately 0.112 microseconds or 33.67 m. In addition, the assertion that $\alpha_{i,0} = 1$ is not

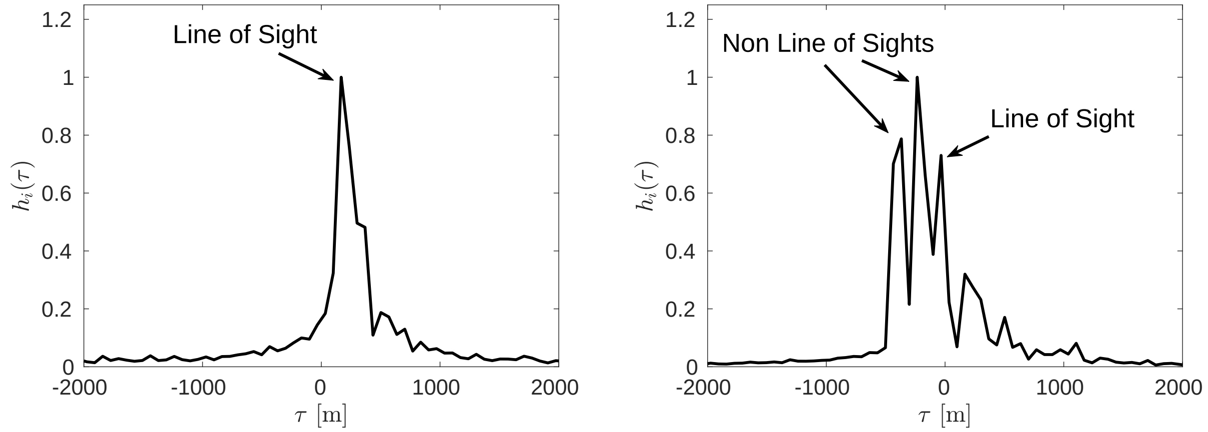

Figure 3.15: Estimated CIR of the Same Data at Different Points in Time

always true. Figure 3.15 shows two CIR estimates from the same data taken at different points in time. This figure shows that the highest peak is not always the LOS peak.

To overcome this drawback, Knutti [52] used a threshold to detect the earliest peak and declare that as the LOS peak. Yet, this does not overcome the issue of resolution. Driusso [31] used a Super Resolution Algorithm (SRA) to estimate the TOA with a high degree of accuracy. The SDR used in this thesis uses this method to refine the initial TOA estimate. In what follows, the SRA-based method will be discussed in detail.

There are multiple SRAs that can be used to solve this problem. The most popular of which are MUltiple SIgnal Classification (MUSIC) and Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT). The ESPRIT method has lower complexity when compared with MUSIC, but it can be less accurate with large Doppler spreads [53]. ESPRIT uses the rotational invariance properties of the subarrays of the subcarriers with respect to each other to estimate e_{TOA} [30, 54]. To do this, the channel length (i.e. the number of delays) L must be estimated. In [55] the Minimum Descriptive Length (MDL) criterion is used to estimate L. The concept of combining ESPRIT and the MDL criterion for TOA estimation for navigation with LTE signals has been used in [22] and [31].

First, the data matrix $\mathbf{X}$ must be constructed with snapshots of the estimated CFR as

$$\mathbf{X} = \begin{bmatrix} \hat{H}'(0) & \hat{H}'(1) & \cdots & \hat{H}'(K-1) \\ \hat{H}'(1) & \hat{H}'(2) & \cdots & \hat{H}'(K) \\ \vdots & \vdots & \cdots & \vdots \\ \hat{H}'(P-1) & \hat{H}'(P) & \cdots & \hat{H}'(M-1) \end{bmatrix} \tag{3.32}$$

where P is the design parameter, $K = M-P+1$, M is the number of subcarriers that contain the CRS, and $\hat{H}^{'}$ is a subset of $\hat{H}$ consisting of only channel estimates for the subcarriers that have the CRS. Next, the channel length L is estimated using the MDL criterion. To do so, the singular value decomposition (SVD) of $\mathbf{X} = \mathbf{U\Sigma V}^{\mathrm{H}}$ must be calculated, where H represents the Hermitian operator, $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices, and $\mathbf{\Sigma}$ is a diagonal matrix with singular values $\sigma_1 \geq \cdots \geq \sigma_P$ on the diagonal. Then, the MDL criterion is calculated as

$$\mathrm{MDL}(\gamma) = -K(P-\gamma)\mathrm{log}\left(\frac{\prod_{l=\gamma}^{P-1}\lambda_l^{1/(P-\gamma)}}{\frac{1}{P-\gamma}\sum_{l=\gamma}^{P-1}\lambda_l}\right) + \frac{1}{2}\gamma(2P-\gamma)\mathrm{log}K, \tag{3.33}$$

$$\text{for} \quad \gamma = 0, \cdots, P-1,$$

$$\hat{L} = \arg\min_{\gamma} \mathrm{MDL}(\gamma). \tag{3.34}$$

Once $\hat{L}$ is found, it is possible to organize the eigenvectors corresponding to the $\hat{L}$ largest eigenvalues as $\mathbf{U}_s = \mathbf{U}[\mathbf{I}_{\hat{L}}\mathbf{0}_{\hat{L}\times(P-\hat{L})}]^{\mathrm{T}}$, where $\mathbf{I}_l$ is an identity matrix of size l, $\mathbf{0}_{l\times p}$ is an l-by-p matrix of zeros. Then, U_1 and U_2 are constructed as

$$\begin{aligned} \mathbf{U}_1 &= [\mathbf{I}_{(P-1)}\mathbf{0}_{P-1\times1}]\mathbf{U}_s, \\ \mathbf{U}_2 &= [\mathbf{0}_{(P-1)\times1}\mathbf{I}_{P-1}]\mathbf{U}_s. \end{aligned} \tag{3.35}$$

Finally, the ESPRIT rotational matrix is constructed as

$$\mathbf{\Psi} = (\mathbf{U}_1^{\mathbf{H}}\mathbf{U}_1)^{-1}\mathbf{U}_1^{\mathbf{H}}\mathbf{U}_2, \tag{3.36}$$

with eigenvalues ψ_l, for $l = 0, \cdots, \hat{L}-1$. The values $\tau_{i,l}$ are then obtained as

$$\tau_{i,l} = -\frac{1}{2\pi T_s \Delta f \Delta_{\text{CRS}}} \arg\{\psi_l\}. \tag{3.37}$$

Since a LOS scenario is assumed, $\tau_{i,0} = 0$, and the normalized estimated TOA error can be found as

$$\hat{e}_{\text{TOA}} = -\min_{l} \tau_{i,l}. \tag{3.38}$$

It should be noted that in non-LOS environments the minimum of the estimated delays will not correspond to the LOS signal. Differentiating between LOS and non-LOS signals is outside of the scope of this thesis.

Frequency Acquisition

To estimate the initial Doppler frequency of the received LTE signal, the difference in phase over time can be measured. To do so, the CRS known at the receiver $S_i(k)$ and the received signal $R_i(k)$ at symbol i are used. Then, define $z(m)$ as [22]

$$\begin{aligned} z(m) &= R_{i+7}(k)R_i^*(k)S_{i+7}^*(k)S_i(k) \\ &= Ce^{j2\pi 7 N_t e_f/N_c}|H_i(k)|^2 + W_{i+7}(k)W_i(k) \\ \text{for} \quad & k \in N_{\text{CRS}}, \quad m = 0, \cdots, M-1. \end{aligned} \tag{3.39}$$

Then, the initial Doppler shift is found as

$$\hat{f}_D = \frac{1}{2\pi T_{\text{slot}}} \Delta\phi, \tag{3.40}$$

where $T_{slot} = 0.5$ ms and

$$\Delta\phi = \arg\left\{\sum_{m=0}^{M-1} z(m)\right\}. \tag{3.41}$$

Finally, the Doppler shift is removed from the received signal as

$$r^{'}(n) = e^{-j2\pi\hat{f}_D nT_s} r(n), \tag{3.42}$$

where $r(n)$ is the received signal samples in the time domain.

After removing the Doppler frequency from the received signal samples, the TOA is corrected by the integer part of $\hat{e}_{\mathrm{TOA}}$. This new TOA is used to start signal tracking. In the case of SSS tracking, signal tracking is normal from here. In the case of CRS tracking, the received samples are converted to the frequency domain as $R^{'}(k) = \mathrm{DFT}\{r^{'}(n)\}$. Then, the fractional part of $\hat{e}_{\mathrm{TOA}}$ is removed as a phase shift in the frequency domain as

$$R^{''}(k) = e^{-j2\pi k \mathrm{Frac}\{\hat{e}_{\mathrm{TOA}}\}/N_c} R^{\prime}(k), \tag{3.43}$$

where $R^{'}(k) = \mathrm{DFT}\{r^{'}(n)\}$.

3.2.4 LTE Signal Tracking

The SDRs used in this thesis track the SSS and the CRS. This subsection discusses the tracking loops used to track these signals.

SSS Tracking

Once the SSS is acquired, signal tracking can begin. In general, the tracking loops used to track the SSS are similar to those used to track GPS, discussed in Section 3.1. The main difference is in code generation. Section 3.2.1 describes the method used in this thesis to generate code replicas. The SSS tracking loop is a modification of the tracking loop shown in Figure 3.6. The most notable changes are the addition of a Frequency Lock Loop (FLL)-assisted PLL and a carrier-aided DLL.

The frequency reuse factor in LTE systems is 1. This results in high interference from neighboring cells. Under high interference, FLLs perform better than PLLs. Yet, PLLs have better measurement accuracy than FLLs. FLL-assisted PLLs combine performance and measurement quality [49]. The main components of an FLL-assisted PLL are a phase discriminator, a phase loop filter, a frequency discriminator, a frequency loop filter, and an NCO. Since the SSS is not modulated with any data, a four-quadrant `atan2` phase discriminator can be used. A number of loop filters can be used in the PLL, but due to high interference, a third-order loop filter is used. The PLL loop filter transfer function is given by [22, 45]

$$F_{\text{PLL}}(s) = 2.4\,\omega_{\text{PLL}} + \frac{1.1\,\omega_{\text{PLL}}^2}{s} + \frac{\omega_{\text{PLL}}^3}{s^2}, \tag{3.44}$$

where ω_{PLL} is the undamped natural frequency. This can be related to the PLL noise-equivalent bandwidth B_{PLL} by $B_{\text{PLL}} = 0.7845\,\omega_{\text{PLL}}$ [45]. The transfer function in equation (3.44) is then discretized. The output of the phase loop filter is the rate of change of the carrier phase in rads/s. The FLL structure is similar to the PLL and DLL discussed in Section 3.1, except the discriminator, does not measure phase error. Instead, the discriminator reports error in frequency. A typical FLL discriminator is shown in equation (3.45)

$$\delta f_k = \frac{\texttt{atan2}(QP_k IP_{k-1} - IP_k QP_{k-1}, IP_k IP_{k-1} + QP_k QP_{k-1})}{T_{\text{PDI}}}, \tag{3.45}$$

where IP_k and QP_k are the in-phase and quadrature prompt correlators at integration period k, respectively. The loop filter used is typically an order lower than the PLL. The transfer function of the second-order loop filter is given by

$$F_{\text{FLL}}(s) = 1.414\,\omega_{\text{FLL}} + \frac{\omega_{\text{FLL}}^2}{s}, \tag{3.46}$$

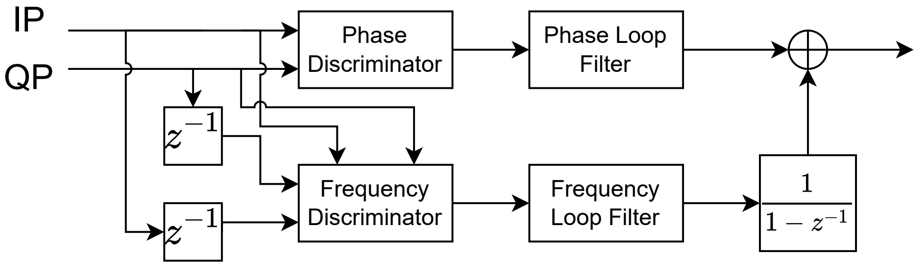

Figure 3.16: Block Diagram of the FLL-Assisted PLL

where ω_{FLL} is the undamped natural frequency. The undamped natural frequency is related to the FLL noise-equivalent bandwidth B_{FLL} by $B_{\text{FLL}} = .53\,\omega_{\text{FLL}}$ [45]. The transfer function in equation (3.46) is then discretized. The output of the frequency loop filter is the rate of change of the carrier frequency, expressed in rads/s^2. Therefore, the output is first integrated and then added to the output of the phase loop filter. Finally, the combined output is integrated by an NCO and used to update the carrier phase of the local replica. Figure 3.16 shows a block diagram of the FLL-assisted PLL.

The SSS tracking loop also uses a carrier-aided DLL. The carrier-aided DLL uses the discriminator shown in equation (3.3). The carrier-aided DLL feeds the output of the FLL-assisted PLL forward to remove dynamic stress error. This is achieved by applying a scale factor to the FLL-assisted PLL output. This works because the Doppler frequency shift on the code is caused by the same dynamics that cause the Doppler frequency shift on the carrier. The scale factor is calculated as

$$SF = \frac{R_c}{f_c}, \tag{3.47}$$

where R_c is the chipping rate and f_c is the carrier center frequency. In the case of the SSS, $R_c = 945$ kHz, and f_c depends on the transmitting eNodeB. In GPS L1 C/A, $R_c = 1.023$ MHz and $f_c = 1575.42$ MHz. Finally, the TOA is updated according to

$$\hat{\tau}_k = \hat{\tau}_{k-1} - \frac{T_{\text{PDI}}}{SPC}(\nu_{\text{DLL}_{k-1}} + SF\nu_{\text{PLL}_{k-1}}), \tag{3.48}$$

where ν_{DLL_k} is the DLL output at integration period k, ν_{PLL_k} at integration period k, and SPC is the number of samples per chip. The newly estimated TOA is used to update the frame timing for the next loop iteration.

CRS Tracking

Unlike the SSS tracking loop, the CRS tracking loop differs from GPS tracking in many ways. For one, the CRS is not contiguous in the frequency domain. Therefore, a time-domain representation does not exist. As a result, the CRS must be tracked in the frequency domain. Figure 3.17 shows the designed CRS tracking loop, based on [22]. First, the Doppler frequency is removed from the incoming signal. Then, the received signal is converted to the frequency domain using a DFT. Since only certain subcarriers contain the CRS, a CRS generator uses system information to produce both the CRS subcarrier indices and the local replica of the CRS. Using the CRS indices, the received CRS is extracted from the received signal in the frequency domain. Since the CRS is found in the frequency domain, the typical carrier and code phase discriminators do not apply here. Instead, alternate frequency-domain versions of these discriminators are used. The carrier phase discriminator is given by [22]

$$\begin{aligned} \delta\phi &= \arg\left[\sum_{m=0}^{M-1} R^{''}(m\Delta_{\text{CRS}}+\kappa)S^{*}(m)\right], \\ \text{for} \quad & m = 0, ..., M-1. \end{aligned} \tag{3.49}$$

After the carrier phase error is extracted, a second-order PLL is used to track the carrier phase. The loop filter has a transfer function given by equation (3.46). The output of the loop filter is integrated by an NCO and used to remove the carrier frequency for the next integration period. The output is also scaled and used in the carrier-aided DLL. The carrier-aided DLL uses a second-order DLL, with a loop filter transfer function given by equation (3.46). The code phase error is extracted by using a frequency-domain code phase discriminator. In conventional DLLs, the code phase error is extracted by using early and

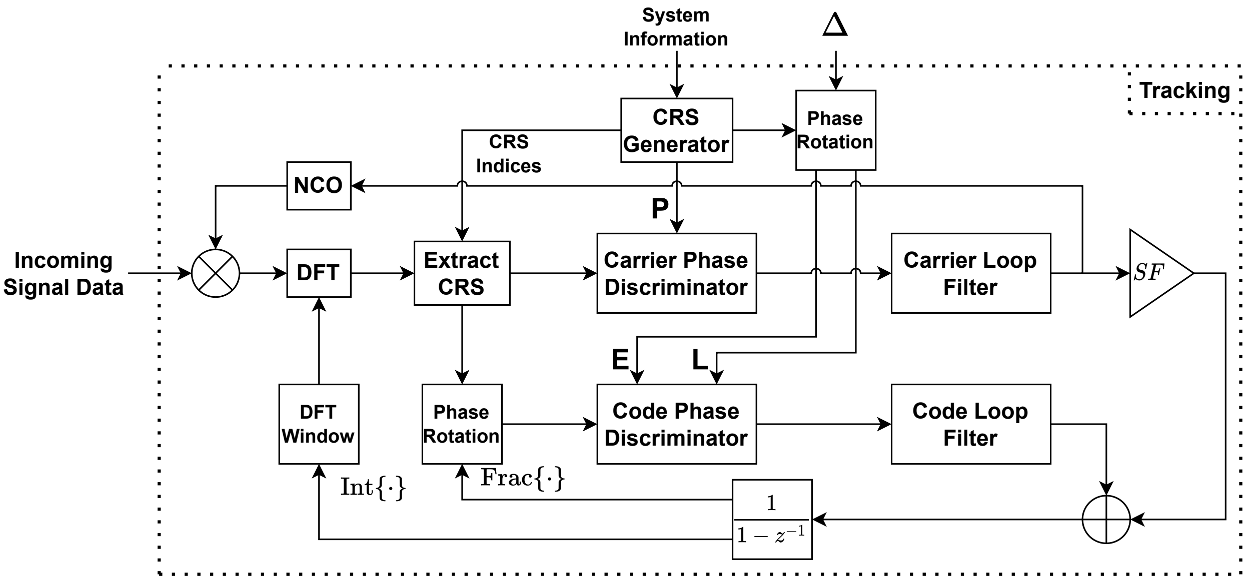

Figure 3.17: Block Diagram of the CRS Tracking Loop

late replicas of the code sequence. To obtain the early and late replicas the code sequence is either advanced or delayed in time, respectively. In the frequency domain, an advance or delay in time is represented by a negative or positive phase rotation, respectively. Therefore, the early and late replicas of the CRS are obtained as

$$\begin{aligned} S_e(m) &= e^{-j2\pi m/M\Delta}S(m), \\ S_l(m) &= e^{j2\pi m/M\Delta}S(m), \\ \text{for} \quad & m = 0,\cdots,M-1, \end{aligned} \tag{3.50}$$

where $0 < \Delta < 1/2$ is the normalized time shift. It is similar to the early-late chip spacing used in conventional tracking loops. The correlators and the frequency domain discriminator function are computed as

$$\begin{aligned}
\mathcal{R}_e &= \sum_{m=0}^{M-1} R^{''}(m\Delta_{\text{CRS}} + \kappa)S_e^*(m), \\
\mathcal{R}_l &= \sum_{m=0}^{M-1} R^{''}(m\Delta_{\text{CRS}} + \kappa)S_l^*(m), \\
e_{DLL} &= |\mathcal{R}_e|^2 - |\mathcal{R}_l|^2. \qquad (3.51)
\end{aligned}$$

The output of the discriminator in equation (3.51) is then normalized by the slope of the S-curve. The slope is defined as [22]

$$k_{\text{DLL}} = \frac{4\pi C cos(\frac{\pi}{2M})}{M(sin(\frac{\pi}{2M}))^3}. \qquad (3.52)$$

The normalized output is then filtered using a loop filter. The loop filter transfer function is the same as in equation (3.46). The output of the loop filter is the rate of change of the symbol start time in samples per second. The output is integrated and used to update the frame start time. The integer part of the updated frame start time is used to control the DFT window, and the fractional part is used to rotate the received CRS replica in the frequency domain.

3.3 RF Positioning

The overall goal of the SDRs discussed in the previous sections is to provide PNT information. To do so, the SDRs track the received signals to extract measurements that are used in a navigation filter. The navigation filter processes the measurements and produces a PNT solution. In this section, PNT solution generation will be discussed, from measurement generation to navigation filter design.

3.3.1 Measurement Generation

To navigate, SDRs make use of observables generated during signal tracking. The primary observables, and the focus of this thesis, are the pseudorange and Doppler shift measurements. This subsection will discuss how each of these measurements are generated, beginning with the pseudorange.

Pseudorange

The pseudorange is the perceived transit time at the receiver. The pseudorange is calculated as

$$\rho = c\, t_{\text{transit}}, \tag{3.53}$$

where c is the speed of light. $t_{\text{transit}} = t_r - t_t$, where t_r and t_t are received time and transmit time, respectively. Therefore, to determine pseudorange, both transmit and receive time must be known. While transmit time can be determined from the GPS data message, the received time is ambiguous. Instead, a so-called delta received time can be extracted from the repeating C/A code. Since the transmit time is regularly scheduled, and the GPS satellite broadcasts clock corrections, the change in transmit time can be considered constant. Hence, we can reasonably approximate the change in transit time as

$$\begin{aligned} \Delta t_{\text{transit}_{k,j}} &= t_{\text{transit}_{k,j}} - t_{\text{transit}_{k-1,j}}, \\ \Delta t_{\text{transit}_{k,j}} &= \Delta t_{r_{k,j}} \end{aligned} \tag{3.54}$$

where $\Delta t_{r_{k,j}}$ is the change in received time at update period k of the C/A code from satellite j. Since the change in transit time is known, an initial transit time is all that is necessary to determine pseudorange. If a reasonably good approximation of the user's position is known at the start of navigation, it can be used to initialize transit time. Otherwise, a reasonable guess must be made. Since the average distance of a GPS satellite to the surface of the earth is $r_j \approx$ 20,000 kilometers, a reasonable guess of transit time can be calculated as

$$t_{\text{transit}_0} = \frac{r_j}{c} \approx 67\text{ms}. \tag{3.55}$$

However, this guess is inherently wrong. Initializing each satellite's transit time with this guess will result in each satellite's pseudorange measurements having a different clock bias, resulting in an unobservable system. To resolve this, only the closest satellite is initialized with equation (3.55). To determine the closest satellite, the decoded data message can be used. Since each satellite reliably transmits at the same time, the satellite with the earliest decoded subframe is initialized with equation (3.55). Then, the transit time of each subsequent satellite is updated using the difference in times of the decoded subframes. This method results in each satellite's pseudoranges having the same initial clock bias, leading to an observable system. Finally, the pseudorange measurement model is written as

$$\rho_j = r_j + c(\delta t_r - \delta t_{s_j}) + T + I + \epsilon + \eta, \tag{3.56}$$

where r_j is the true range from the j-th satellite to the receiver, δt_r is the receiver clock bias, δt_{s_j} is the j-th satellite clock bias, T is the delay induced by the troposphere, I is the delay induced by the ionosphere, ϵ are other errors due to unmodeled effects, η is AWGN. To obtain a more precise solution, an ionospheric model or a dual-frequency approach could be used to remove the error due to the ionosphere [56, 57]. The tropospheric error is usually small but can also be modeled [58]. These methods are outside the scope of this thesis. For simplicity, equation (3.56) can be rewritten as

$$\rho_j = r_j + c\delta t_r + \eta, \tag{3.57}$$

since the unmodeled effects are relatively small, and the satellite clock bias is provided by the ephemeris.

In the case of LTE, equation (3.57) will still apply. The LTE data message does not provide a clock term like GPS, but the eNodeB clock bias can be estimated offline. The other difference lies in the fact that an initial transit time will not be used for LTE, in this thesis. Instead, an initial user position will be assumed and used to determine the initial pseudorange. Then, the delta pseudorange can be constructed in the same way.

Doppler Shift

The Doppler shift is the perceived change in frequency due to receiver and transmitter dynamics. It is commonly used to estimate the velocity of the receiver. In addition, it can be integrated to generate the accurate carrier phase measurement. The initial Doppler shift measurement is generated during acquisition for both GPS and LTE. Then, it is updated by the output of the PLL or FLL-assisted PLL for both GPS and LTE, respectively. The Doppler shift equation is given by

$$f_{D_j} = -\dot{\rho}_j / \lambda_c, \tag{3.58}$$

where λ_c is the wavelength of the carrier, and $\dot{\rho}$ is the pseudorange rate of the j-th emitter relative to the receiver. The pseudorange rate is defined as

$$\dot{\rho}_j = \dot{r_j} + c(\dot{\delta t_r} - \dot{\delta t_{s_j}}) + \nu, \tag{3.59}$$

where $\dot{r_j}$ is the range rate of the j-th emitter relative to the receiver, $\dot{\delta t_r}$ is the clock drift of the receiver, $\dot{\delta t_{s_j}}$ is the clock drift of the j-th emitter, and ν is AWGN. In the case of GPS, the transmitter clock drift is considered to be negligible. Again, the clock drift of the transmitting eNodeB is estimated offline. Hence, equation (3.59) can be rewritten as

$$\dot{\rho}_j = \dot{r}_j + c\,\delta\dot{t}_r + \nu. \tag{3.60}$$

3.3.2 Navigation Filter Design

Finally, a navigation filter can be designed considering the measurement models described in equations (3.57) and (3.59). First, the estimated states must be defined. The primary states of interest to the user are the position and velocity states. However, just estimating the position and velocity would lead to large errors due to unmodeled clock terms. As a result, most designs elect to estimate position, clock bias, velocity, and clock drift. The final state vector is formed as

$$\mathbf{x} = \begin{bmatrix} \mathbf{r} & c\delta t_r & \dot{\mathbf{r}} & c\delta \dot{t}_r \end{bmatrix}^{\mathrm{T}}, \tag{3.61}$$

where $\mathbf{r}$ represents the vector of x, y, and z positions and $\dot{\mathbf{r}}$ represents the vector of x, y, and z velocities. Then, the state vector can be related to equations (3.57) and (3.60) by

$$\begin{aligned} h(\mathbf{x})_{\rho_j} &= \sqrt{(x_j - x_r)^2 + (y_j - y_r)^2 + (z_j - z_r)^2} + c\delta t_r + 0 \times \dot{\mathbf{r}} + 0 \times c\delta \dot{t}_r, \\ h(\mathbf{x})_{\dot{\rho}_j} &= (\dot{x}_j - \dot{x}_r)a_{x,j} + (\dot{y}_j - \dot{y}_r)a_{y,j} + (\dot{z}_j - \dot{z}_r)a_{z,j} + c\delta \dot{t}_r + 0 \times \mathbf{r} + 0 \times c\delta t_r, \end{aligned} \tag{3.62}$$

, where x, y, and z are components of position and the subscripts j and r, represent emitter and receiver, respectively, the $\dot{}$ represents the time-derivative, and $a_{x,j}$, $a_{y,j}$, and $a_{z,j}$ represent the x, y, and z unit vectors from the receiver to the j-th emitter, respectively. These observation equations are nonlinear, precluding them from being used in a traditional least squares (LS) estimator. Instead, a Non-linear LS (NLS) estimator is used. The NLS estimator uses the linearized form of equation (3.62) for each of the m emitters. The resultant linearizations are combined to form a $2m \times 8$ observation matrix. The observation matrix is constructed as

$$\mathbf{H} = \begin{bmatrix} a_{x,1} & a_{y,1} & a_{z,1} & 1 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x,m} & a_{y,m} & a_{z,m} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{x,1} & a_{y,1} & a_{z,1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & a_{x,m} & a_{y,m} & a_{z,m} & 1 \end{bmatrix}. \tag{3.63}$$

Instead of using the pseudorange and Doppler shift measurements directly, NLS uses the so-called residuals of those measurements. That is the difference between a locally predicted and a receiver-generated measurement. With this in mind, a measurement residual vector can be constructed as

$$\mathbf{z} = \begin{bmatrix} \hat{\rho}_1 - \tilde{\rho}_1 & \cdots & \hat{\rho}_m - \tilde{\rho}_m & \hat{\dot{\rho}}_1 - \tilde{\dot{\rho}}_1 & \cdots & \hat{\dot{\rho}}_m - \tilde{\dot{\rho}}_m \end{bmatrix}^{\mathrm{T}}, \tag{3.64}$$

where $\hat{}$ and $\tilde{}$ represent the receiver estimate and the measurement, respectively. Finally, the states are updated according to

$$\mathbf{x}^{+} = \mathbf{x}^{-} + (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{z}. \tag{3.65}$$

It should be noted that, to update the states as such, some *a priori* estimates of the states must be known. If the location of the receiver is completely unknown, an initial estimate of an 8 × 1 vector of zeros is sufficient. Since this estimate is relatively poor, equation (3.65) must be iterated until $(\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{z}$ is sufficiently small. If the initial estimate is reasonable, iteration is not necessary. With regard to LTE-only navigation, the initialization must be better to assure convergence.

Chapter 4

Vector Tracking Loops

The tracking loops discussed in the previous chapter are integral to the function of navigation receivers. They extract the measurements necessary to perform positioning and timing. Yet, they use a naive approach to track the receiver and satellite dynamics. The noise equivalent bandwidth is adjusted prior to use to capture these dynamics. Without some *a priori* knowledge of the receiver and satellite dynamics, the selected bandwidth will not be optimal. Some dynamics will be neglected, or unnecessary noise will be allowed into the system.

A way to resolve this could be to use an adaptive Kalman filter to select optimal bandwidths such as in [59]. However, this method still makes no assumptions about the receiver or emitter motion. Instead, the bandwidths are adaptively estimated using measurement residuals from the discriminators. While this is an improvement over fixed-bandwidth methods, each channel is still tracked individually. As a result, very low-powered channels may still be lost since their discriminators report mostly noise.

To improve upon this further, an estimate of the receiver and satellite dynamics could be used at each integration period to compute an adaptive bandwidth. To do so, a navigation solution must be computed at each integration period, assuming at least four emitters are present. With regard to GPS, this is almost always the case. The advantage of this approach is that it combines the adaptive bandwidth elements of [59] with the filtering effect provided by navigation. That is, as long as there are four channels of fairly good quality, even very low-powered channels can be tracked.

This chapter discusses the vector tracking algorithms used in this thesis. In particular, Section 4.1 describes the Vector Delay/Frequency Lock Loop (VDFLL) algorithm. Section 4.2 discusses the architecture of the combined GPS and LTE VDFLL algorithm.

4.1 Vector Delay/Frequency Lock Loop

The classical vector tracking algorithm was first proposed by [8]. This algorithm was referred to as the vector delay lock loop (VDLL). In the VDLL, the code frequency is tracked using an Extended Kalman Filter (EKF). The EKF estimates the navigation solution and updates the tracking loops, improving tracking performance. In [60], Pany explored tracking the code and carrier frequency together. This algorithm was later referred to as the vector delay/frequency lock loop (VDFLL). This algorithm showed marked improvement over scalar tracking methods when tracking weak GPS signals under high dynamic stress [5, 10, 61]. More recently, it has been shown to increase resilience to multipath fading as well [62, 63]. In the following sections, the algorithm will be described. For a more thorough treatment of vector tracking algorithms, the reader is encouraged to refer to [6] or [12]. A block diagram of the VDFLL algorithm is provided in Figure 4.1, showing how the signal processing is unified in the navigation processor.

4.1.1 Extended Kalman Filter

To begin tracking with a VDFLL, the receiver is assumed to have an *a priori* estimate of the receiver's states described by

$$\hat{\mathbf{x}}_k^- = \begin{bmatrix} x & \dot{x} & y & \dot{y} & z & \dot{z} & c\delta t_r & c\dot{\delta t_r} \end{bmatrix}^T , \tag{4.1}$$

with covariance $\hat{\mathbf{P}}_k^-$, where x, y, and z are the receiver's position states in meters, $\dot{x}$, $\dot{y}$, and $\dot{z}$ are the receiver's velocity states in meters per second, δt_r is the receiver clock bias in seconds, $\dot{\delta t_r}$ is the receiver clock drift in seconds per second, and c is the speed of light in meters per

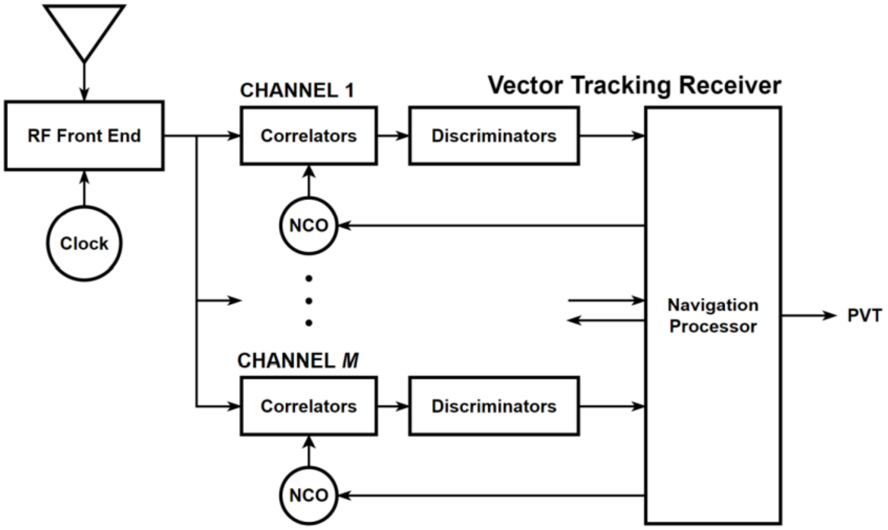

Figure 4.1: The classical VDFLL block diagram.

second. It should be noted that the state vector in equation (4.1) is different than the one in (3.61). This is to illustrate that the order of the state vector itself does not matter. Instead, the order of the state vector will change the formulation of the state transition matrix and observation matrix. The state vector in (4.1) is selected for the VDFLL because it simplifies the state transition matrix in the EKF. Then, the dynamic model of the receiver is described by

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k, \tag{4.2}$$

where $\mathbf{F}$ is the state transition matrix given by

$$\mathbf{F} = \begin{bmatrix} \mathbf{A} & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{A} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{A} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{A} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \mathbf{O} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.3}$$

and $\mathbf{w}_k$ is the process noise vector drawn from a zero mean multivariate Gaussian distribution with covariance $\mathbf{Q}$. $\mathbf{Q}$ is written as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_x & \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{Q}_y & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{Q}_z & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \mathbf{Q}_{clk} \end{bmatrix}, \tag{4.4}$$

where

$$\mathbf{Q}_n = \begin{bmatrix} \sigma_n^2 \frac{T^3}{3} & \sigma_n^2 \frac{T^2}{2} \\ \sigma_n^2 \frac{T^2}{2} & \sigma_n^2 T \end{bmatrix}, \quad \mathbf{Q}_{clk} = \begin{bmatrix} \sigma_b^2 T + \sigma_{br}^2 T^2 + \sigma_r^2 \frac{T^3}{3} & \sigma_{br}^2 T + \sigma_r^2 \frac{T^2}{2} \\ \sigma_{br}^2 T + \sigma_r^2 \frac{T^2}{2} & \frac{\sigma_b^2}{T} + \sigma_{br}^2 + \frac{4}{3}\sigma_r^2 T \end{bmatrix}, \tag{4.5}$$

where σ_n^2 describes the variance of the noise processes that drive the x, y, or z velocity errors in $\mathrm{m}^2/\mathrm{s}^3$, σ_b^2 is the variance of the noise process that drives the clock phase error in m^2/s, σ_r^2 is the variance of the noise process that drives the clock frequency error in $\mathrm{m}^2/\mathrm{s}^3$, and σ_{br}^2 is the covariance of the noise processes that drives the clock phase and frequency error in $\mathrm{m}^2/\mathrm{s}^2$. T is the integration period and it is used to scale the variances to the time step in the EKF. The σ_n^2 terms are often considered tuning parameters and are similar to noise bandwidths in scalar tracking loops, but they are usually selected based on some assumption of the expected receiver dynamics. A guideline for the selection of σ_n^2 may be [64]

$$\sigma_n = \frac{v_n}{\sqrt{T}} \tag{4.6}$$

where v_n is the largest instantaneous velocity the receiver is expected to experience in the n direction. The clock terms σ_b^2 and σ_{br}^2 are selected according to the quality of the receiver clock. Some nominal values may be found in [65]. Finally, the Kalman filter time update may be written as

$$\hat{\mathbf{x}}_{k+1}^{-} = \mathbf{F}\hat{\mathbf{x}}_{k}^{-} \tag{4.7}$$

$$\mathbf{P}_{k+1}^{\hat{-}} = \mathbf{F}\mathbf{P}_{k}^{\hat{-}}\mathbf{F}^{T} + \mathbf{Q} \tag{4.8}$$

Then, the measurement model is described by

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \tag{4.9}$$

where $\mathbf{y}_k$ is the observation vector, h is a nonlinear function that maps the states to the measurements, given in equation (3.62), and $\mathbf{v}_k$ is the measurement noise vector that is drawn from a zero mean multivariate Gaussian distribution with covariance $\mathbf{R}$. It is important to note that for an EKF, equation (4.9) must be linearized. The linearized form uses the Jacobian of h, H, and measurement residuals $\mathbf{z}$, where

$$\mathbf{H} = \begin{bmatrix} a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{x,m} & 0 & a_{y,m} & 0 & a_{z,m} & 0 & 1 & 0 \\ 0 & a_{x,1} & 0 & a_{y,1} & 0 & a_{z,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & a_{x,m} & 0 & a_{y,m} & 0 & a_{z,m} & 0 & 1 \end{bmatrix}, \tag{4.10}$$

$$\mathbf{z} = \begin{bmatrix} \delta\rho_1 & \cdots & \delta\rho_m & \delta\dot{\rho}_1 & \cdots & \delta\dot{\rho}_m \end{bmatrix}, \tag{4.11}$$

and a_{n_j} is the unit vector to the j-th emitter in the x, y, or z direction, $\delta\rho$ is the pseudorange residual in m, $\delta\dot{\rho}$ is the pseudorange rate residual in m/s. The residuals are obtained from conventional code and frequency discriminators within the tracking loop and are discussed in the following section. Once the Jacobian and measurement residual vector are obtained, the Kalman filter measurement update is computed as

$$\begin{aligned}\mathbf{K} &= \hat{\mathbf{P}}^{-}\mathbf{H}^{T}(\mathbf{H}\hat{\mathbf{P}}^{-}\mathbf{H}^{T}+\mathbf{R}), \\ \hat{\mathbf{x}}^{+}_{k+1} &= \hat{\mathbf{x}}^{-}_{k+1}+\mathbf{K}(\mathbf{z}-\mathbf{H}\hat{\mathbf{x}}^{-}_{k+1}), \\ \mathbf{P}^{\hat{+}}_{k+1} &= (\mathbf{I}-\mathbf{KH})\mathbf{P}^{\hat{-}}_{k+1}(\mathbf{I}-\mathbf{KH})^{T}+\mathbf{KRK}^{T},\end{aligned} \tag{4.12}$$

where **R** is the measurement covariance matrix; it will be discussed in the following section.

To illustrate the advantages of vector tracking, Figure (4.2) compares vector and scalar tracking algorithms when subjected to a partial outage situation. In the simulations all but two GPS signals are degraded. The vector receiver adaptively estimates the C/N_0 and the tracking can continue using the two high-quality signals. The outage ends at approximately 75s and the VDFLL continues to produce high-quality replicas of the degraded signals. In the scalar case, the degraded signals must be reacquired.

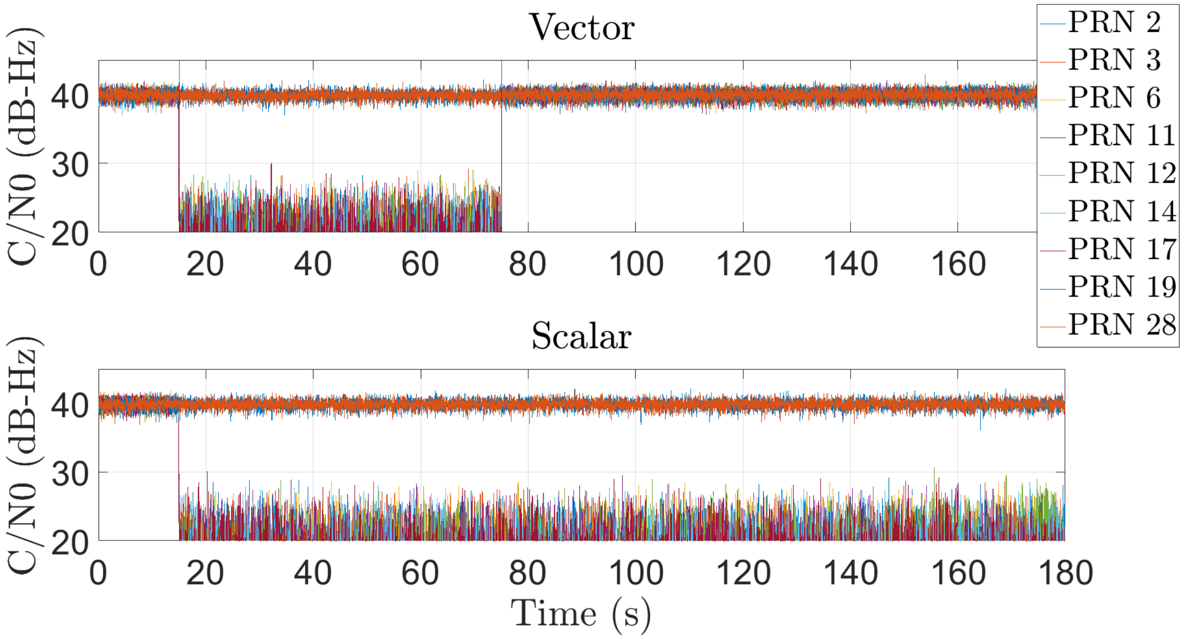

Figure 4.2: Vector and scalar tracking algorithms, subjected to a 60-second partial outage, beginning at $t = 15s$.

4.2 GPS and LTE VDFLL

This section discusses the architecture of the combined VDFLL and the incurred differences when compared with the traditional GPS VDFLL. Since LTE was not designed with navigation in mind, certain changes to the original architecture must be made. These changes include differences in eNodeB timing state extraction, measurement residuals and covariance calculation, and Carrier-to-Noise ratio (C/N_0) estimation.

4.2.1 Combined Architecture

The combined GPS/LTE VDFLL architecture is similar to the one presented in Figure 4.1 with one key hierarchical change. Figure 4.3 shows the modified VDFLL architecture block diagram. For both GPS and LTE a scalar tracking segment is performed first. In the case of GPS, this scalar tracking estimate is used to determine an initial PVT solution and associated covariance to begin the vector tracking. Additionally, it supplies the vector tracking loop with the ephemeris of the transmitting GPS satellites. The scalar tracking of the LTE signals does not produce a position solution. Instead, the PVT solution from the GPS scalar tracking is used as an input such that the LTE scalar tracking loop produces measurements of the transmitter clock bias and drift only. The LTE scalar tracking is performed according to the CRS receiver discussed in 3. For more detail, an in-depth performance analysis of the scalar receiver is provided in [22]. It is worth noting that the 'PreProcessing' block shown in Figure 4.3 is not necessarily required to be a preprocessing step. Indeed, in most GPS VDFLL configurations scalar tracking is performed first to initialize vector tracking. This could be done sequentially with little latency. With regard to LTE, extracting the timing states of the transmitters must be done offline. However, this does not preclude this system from functioning in real time. A separate, fixed-location base station with an onboard GPS receiver could be used to estimate the eNodeB clock states and send them to the receiver using the combined VDFLL. In this thesis, this step will be performed as an initialization, instead.

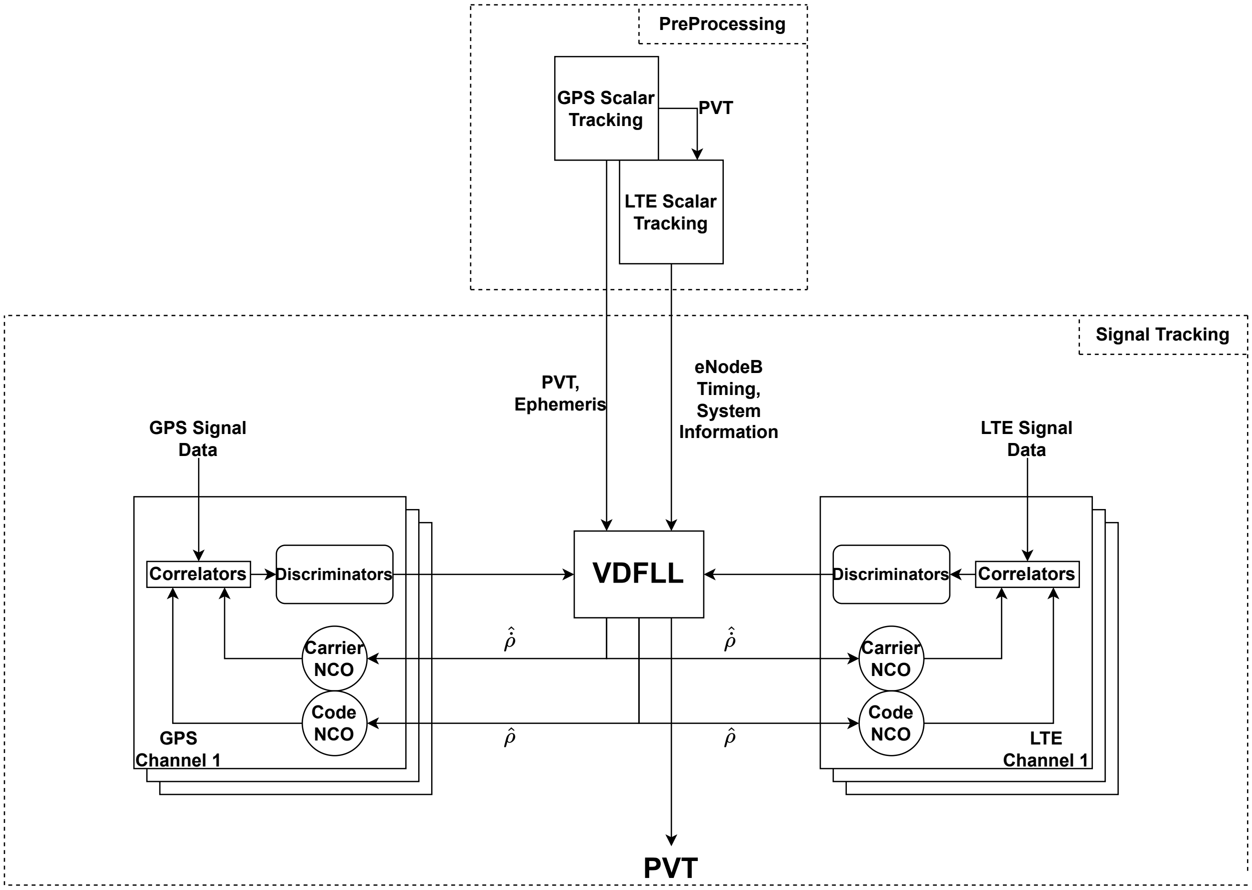

Figure 4.3: The full GPS and LTE VDFLL architecture block diagram.

4.2.2 Measurement Residual Calculation

Traditionally, the scalar tracking receiver will generate measurements of the pseudorange, $\tilde{\rho_j}$. The receiver will then estimate its pseudorange to the j-th emitter, $\hat{\rho_j}$. The difference, $\delta\rho_j = \hat{\rho_j} - \tilde{\rho_j}$, is referred to as the measurement residual. This is then used in a non-linear estimator to determine the PVT states. In the VDFLL, direct measurements of the pseudoranges and pseudorange rates are not available. Instead, the residuals themselves are observable. Consider the output of equations (3.3) and (3.45), $\delta\tau$ and δf. These discriminators output errors in the code phase and the Doppler frequency of the tracked signal. These errors can then be scaled to represent the pseudorange and pseudorange rate residuals. These are computed as

$$\begin{aligned} \delta\rho &= \lambda_{\text{chip}}\delta\tau, \\ \delta\dot{\rho} &= -\lambda_{\text{carrier}}\delta f, \end{aligned} \tag{4.13}$$

where λ represents the code and carrier wavelengths in meters. For the GPS L1 C/A signal, the wavelengths are $\lambda_{\text{chip}} = 293.05$ m and $\lambda_{\text{carrier}} = .1905$ m, respectively. For the LTE signal, the wavelength of the SSS is $\lambda_{\text{chip}} = 317.46$ m. The wavelength of the CRS and the carrier both depend on the transmitting eNodeB.

Measurement Covariance Determination

The measurement covariance matrix, $\mathbf{R}$, is typically determined by C/N_0. In [66], Crane describes a function to compute range and range rate residual covariance as a function of C/N_0 for GPS. In most vector tracking implementations, the C/N_0 is estimated online and the measurement covariance matrix $\mathbf{R}$ is updated. This, in part, is what makes the vector tracking algorithm robust. For GPS, the measurement covariance for the pseudorange and pseudorange rate residual are described by [66]

$$\sigma^2_{\delta\rho,\text{GPS}} = \frac{\lambda^2_{\text{chip}}}{2T^2_{\text{PDI}}(C/N_0)^2} + \frac{\lambda^2_{\text{chip}}}{4T_{\text{PDI}}(C/N_0)},$$
$$\sigma^2_{\delta\dot{\rho},\text{GPS}} = K\left(\frac{\lambda^2_{\text{carrier}}}{\pi^2 T^2_{\text{PDI}}}\right)\left(\frac{2}{T^2_{\text{PDI}}(C/N_0)^2} + \frac{2}{T_{\text{PDI}}(C/N_0)}\right),$$

where C/N_0 is the estimated carrier-to-noise power density ratio, and K is a scaling constant that can be used to increase the pseudorange rate residual variance in high C/N_0 conditions. For LTE signals, a different method of computing the residual variances must be used. With regard to the SSS, this is because the autocorrelation function resembles a sinc function instead of a triangle. As for the CRS, the pseudorange residual is computed directly in the frequency domain, warranting a new variance calculation. The calculations are derived in [22] and are reproduced in equation (4.14) for clarity.

$$\begin{aligned}\sigma^2_{\delta\rho,\text{SSS}} &= \frac{\lambda^2_{\text{chip}}}{k_d^2}\left(\frac{2}{T^2_{\text{PDI}}(C/N_0)^2} + \frac{4\,\text{sinc}(\Delta/2)}{T_{\text{PDI}}(C/N_0)}\right)\\ \sigma^2_{\delta\rho,\text{CRS}} &= \frac{\lambda^2_{\text{sample}}}{Nk_d^2}\frac{\pi^2}{64MT_{\text{symbol}}(C/N_0)}\end{aligned} \tag{4.14}$$

k_d refers to the slope of the code phase discriminator S-curve, which is different for the SSS and the CRS. λ_{sample} refers to the width of a sample in meters (Table 4.1). M corresponds to the number of subcarriers containing the CRS. N is the number of OFDMA symbols integrated. T_{symbol} is the OFDMA symbol period. T_{PDI} is the predetection integration period. Equation (4.14) holds if $M >> 1$ and $T_{\text{symbol}} \times C/N_0 = SNR >> 1$.

Table 4.1: Available bandwidths, sampling rates, and corresponding sample widths [69].

Total Bandwidth (MHz)	**Sampling Rate (MHz)**	**Sample Width (m)**
1.4	1.92	156.14
3	3.84	78.07
5	7.68	39.04
10	15.36	19.51
15	23.04	13.01
20	30.72	9.76

The pseudorange rate residual variances are computed similarly, but λ_{carrier} is changed to match the center frequency of the broadcasting eNodeB. The pseudorange rate residual variances are determined as

$$\begin{aligned}\sigma^2_{\delta\dot{\rho},\text{SSS}} &= K\left(\frac{\lambda^2_{\text{carrier}}}{\pi^2 T^2_{\text{PDI}}}\right)\left(\frac{2}{T^2_{\text{PDI}}(C/N_0)^2}+\frac{2}{T_{\text{PDI}}(C/N_0)}\right),\\ \sigma^2_{\delta\dot{\rho},\text{CRS}} &= \left(\frac{\lambda^2_{\text{carrier}}}{(N/2)\pi^2 T^2_{\text{PDI}}}\right)\left(\frac{1}{MT_{\text{symbol}}(C/N_0)}+\frac{1}{2M^2T^2_{\text{symbol}}(C/N_0)^2}\right).\end{aligned} \tag{4.15}$$

Finally, $\mathbf{R}$ is constructed as

$$\mathbf{R} = \begin{bmatrix} \sigma^2_{\delta\rho,0} & & & & & \\ & \ddots & & & & \\ & & \sigma^2_{\delta\rho,m} & & & \\ & & & \sigma^2_{\delta\dot{\rho},0} & & \\ & & & & \ddots & \\ & & & & & \sigma^2_{\delta\dot{\rho},m} \end{bmatrix}, \tag{4.16}$$

where the off-diagonal terms are 0 and m corresponds to the number of emitters. These variances are only valid if C/N_0 is known by the receiver. In most cases, C/N_0 must be estimated online. To do so, a Kalman filter-based estimator can be used [67]. In this thesis, a moving-average type estimator is used. For GPS, the C/N_0 can be estimated using the amplitude of the autocorrelation function A. Its square is determined by

$$A^2 + 4\sigma^2_\eta = (IE + IL)^2 + (QE + QL)^2 \tag{4.17}$$

where IE, IL, QE, and QL are the in-phase and quadrature early and late correlator outputs, respectively. Then, using the estimated amplitude the C/N_0 is given by

$$C/N_{0,\text{GPS}} = 10\log_{10}\left(\frac{\hat{A}^2 - 4\hat{\sigma}^2_\eta}{2T_{\text{PDI}}\sigma^2_\eta}\right), \tag{4.18}$$

where $\hat{\sigma}_\eta^2$ is the estimate of the correlator noise variance, and $\hat{A}^2 = (IE+IL)^2+(QE+QL)^2$. It is typically obtained by correlating the received GPS signal against a code replica of a different PRN. Since the cross-correlation of different PRNs is 0 everywhere, the output is just noise. Hence, early, late, and prompt in-phase and quadrature correlator outputs are produced using code replicas with unused PRNs, creating the so-called 'noise correlators'. Then, the variance of the resulting correlators is used as $\hat{\sigma}_\eta^2$.

With regard to LTE, (4.18) must be modified according to the autocorrelation functions of the SSS and the CRS. For the SSS, equation the amplitude equation can be rewritten as

$$4\,\text{sinc}^2(\Delta/2)A^2 + 4\sigma_\eta^2 = (IE+IL)^2 + (QE+QL)^2, \tag{4.19}$$

assuming no code phase error. Therefore, the measurement of amplitude is scaled by $4\,\text{sinc}^2(\Delta/2)$. To correct this, this estimate can be scaled down by the same factor. This results in the following changes to (4.18)

$$C/N_{0,\text{SSS}} = 10\log_{10}\left(\frac{\hat{A}^2 - \frac{\hat{\sigma}_\eta^2}{\text{sinc}^2(\Delta/2)}}{2T_{\text{PDI}}\sigma_\eta^2}\right), \tag{4.20}$$

assuming $\hat{A}^2 = \frac{(IE+IL)^2+(QE+QL)^2}{4\text{sinc}^2(\Delta/2)}$. Similar to GPS, an unused SSS sequence can be used to determine the noise correlators used to calculate the noise variance estimate $\hat{\sigma}_\eta^2$.

Signal-to-Noise Ratio (SNR) estimation of an OFMDA signal using pilot sequences, such as the CRS, has been discussed thoroughly in the literature. Popular approaches use the CFR to estimate the signal and noise subspaces [68]. The signal power A is estimated from the signal subspace and the noise power $\hat{\sigma}_\eta^2$ from the noise subspace. While accurate, these methods are computationally intensive. As a result, they do not lend themselves well as online estimators. Instead, a biased estimator like in equations (4.17) and (4.19) is used. Again, the noise power estimate $\hat{\sigma}_\eta^2$ can be determined using an unused CRS sequence. The biased estimator is developed as

$$2M^2 \left(\frac{\sin(\pi\Delta)}{M \sin(\pi\Delta/M)} \right)^2 \left(A^2 + 4\sigma_\eta^2 \right) = |\mathcal{R}_e|^2 + |\mathcal{R}_l|^2. \tag{4.21}$$

It follows that

$$A^2 + 4\sigma_\eta^2 = \frac{|\mathcal{R}_e|^2 + |\mathcal{R}_l|^2}{2M^2 \left(\frac{\sin(\pi\Delta)}{M \sin(\pi\Delta/M)} \right)^2}, \tag{4.22}$$

and

$$C/N_0 = 10\log_{10} \left(\frac{\hat{A}^2 - 4\hat{\sigma}_\eta^2}{2\sigma_\eta^2 T_{\text{symbol}}} \right), \tag{4.23}$$

where $\hat{A}^2 = \frac{|\mathcal{R}_e|^2 + |\mathcal{R}_l|^2}{2M^2 \left(\frac{\sin(\pi\Delta)}{M \sin(\pi\Delta/M)} \right)^2}$. Finally, the receiver's states are updated using the traditional Kalman filter measurement update shown in (4.13). The resulting states are representative of the receiver's states at the end of the integration period.

To prepare the code and carrier replicas for the next integration period a temporary time update is performed. Then, the states are passed through the non-linear measurement model to produce the next estimates of the pseudoranges and pseudorange rates. These estimates are used to generate the code and carrier frequencies used to update the local replicas. These updates are determined by

$$\begin{aligned} f_{\text{code}_{k+1}} &= f_{\text{code}_k} - \frac{\hat{\rho}_{k+1} - \hat{\rho}_k}{\lambda_{\text{chip}} T_{\text{PDI}}}, \\ f_{\text{carrier}_{k+1}} &= \frac{\hat{\dot{\rho}}_{k+1}}{-\lambda_{\text{carrier}}}. \end{aligned} \tag{4.24}$$

In the GPS- and SSS-based tracking loops, these updates are outputted in units of chips per second and Hz, respectively. These values are integrated by code and carrier NCOs as normal. In the CRS-based tracking loop, the f_{code} output is in units of samples per second. It cannot be integrated by a code NCO in the same way. Instead, the output is accumulated and saved as

$$\Sigma_{k+1} = \Sigma_k + f_{\text{code}_{k+1}} T_{\text{PDI}}. \tag{4.25}$$

The integer part of the accumulation is used to update the frame start time as

$$\tau_{k+1} = \tau_k + \mathrm{int}\{\Sigma_{k+1}\}, \tag{4.26}$$

and the fractional part is used to rotate the received CRS symbols as

$$R'''(k) = e^{-j2\pi k \mathrm{Frac}\{\Sigma_{k+1}\}/N_c} R''(k). \tag{4.27}$$

Note that $R''(k)$ is found after acquisition, and therefore, we may set $\Sigma_0 = \mathrm{Frac}\{\hat{e}_{\mathrm{TOA}}\}$. Once the integer part of the accumulation is used to update the frame start time, it is removed from the total accumulation. For convenience, both methods are referred to as an NCO in Figure 4.3.

Chapter 5

Simulation and Experimental Results

In this chapter, the performance of the combined GPS/LTE VDFLL is evaluated. The performance is evaluated in static and dynamic scenarios in both simulation and experimentation. The simulations will be discussed first.

Real GPS ephemeris data are used to propagate realistic GPS satellite trajectories for the simulations. The location of 1 real eNodeB is used for the simulation, and 3 additional eNodeBs are also simulated to show the performance in environments where more than 1 eNodeB are available. In the simulations, the receiver is evaluated based on its probability of tracking the GPS and LTE signals and its Root Mean Squared Error (RMSE) in code phase and carrier frequency tracking over a range of C/N_0. Also, the probability of tracking the signals is analyzed. In the simulations, the receiver is subjected to AWGN channels first. Then, the LTE signals will be subjected to multipath channels. The results of the combined receiver are compared to GPS-only receivers (scalar and vector), and LTE-only receivers (scalar). Again, the GPS-only receivers are only subjected to AWGN channels, while the LTE-only receivers will be subjected to AWGN and multipath channels. Finally, the combined VDFLL will be subjected to periodic GPS outages. The LTE signals will be kept at high C/N_0 but will be subjected to multipath.

Finally, the combined GPS/LTE VDFLL will be tested experimentally. Real signal data are recorded near the Thomas Walter MRI research building in Auburn, Alabama, and in downtown Atlanta, Georgia. Static and dynamic data are collected. A Honeywell eTalin is used as a reference. The eTalin provides a high-grade GPS/INS solution that is used as ground truth for the experiments. The positioning results of the combined VDFLL are compared with a GPS-only VDFLL. Subsection 5.1.2 describes the simulation methodology that was presented in [70]. Some of the results in Subsection 5.2 were shown in [71].

5.1 Simulations

To evaluate the performance of the combined VDFLL, simulations are performed across a wide variety of conditions. First, the GPS and LTE signals are subjected to AWGN channels. The code phase and carrier frequency tracking RMSE are analyzed. These are compared with vector and scalar standalone GPS receivers and scalar standalone LTE receivers, respectively. To perform this comparison, the information-theoretic lower bound on performance is derived for the combined vector receiver using the Discrete Algebraic Riccati Equation (DARE). It is compared to the lower bound on the performance of the standalone GPS VDFLL, derived in the same way. The lower bound on the performance of the scalar receivers is determined using loop theory [45]. Then, the theoretical bounds are verified by Monte Carlo simulation. In addition, the Monte Carlo simulations show the effect of errors that cannot be captured by the theoretical calculations, such as errors due to emitter localization. Next, Monte Carlo simulations are performed to evaluate the performance of the combined VDFLL when the LTE signals are subjected to multipath channels. Monte Carlo simulations are used because the effects of multipath cannot be modeled by DARE. Finally, Monte Carlo simulations are performed to determine the survival rate of GPS signals when using the combined VDFLL under GPS outage conditions, while LTE signals are affected by multipath and eNodeB localization errors.

5.1.1 Simulation Setup

All simulations conducted in this thesis take place in the environment shown in Figure 5.1. The simulated receiver is initially placed on the roof of the Thomas Walter MRI building at Auburn University. Real GPS ephemeris data are used to propagate realistic GPS satellite trajectories for the simulations. Two sets of ephemeris data are used. Both sets were collected

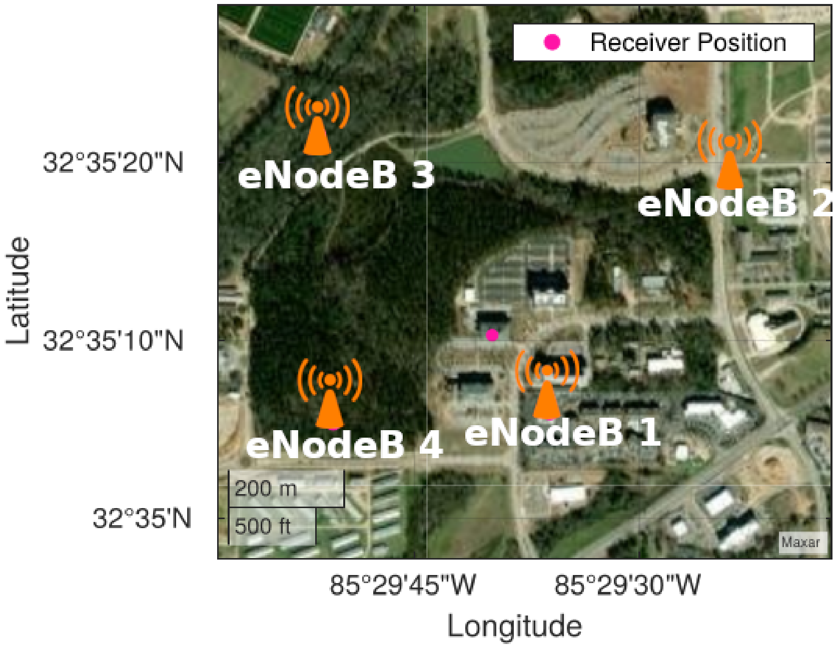

Figure 5.1: Simulation Environment

at the MRI building. One set consists of 4 GPS satellites and the other consists of 5 GPS satellites. The initial skyplots for each set are shown in Figure 5.2. Each set is termed 'static' or 'dynamic' to denote the simulation scenario in which they are used. The specific scenarios will be discussed further, later. In addition to the GPS satellites, eNodeBs are simulated. Their positions are held constant throughout the simulations and are shown in Figure 5.1. The eNodeBs are given center frequencies according to Table 5.1. These center frequencies are selected based on real center frequencies found on eNodeBs in the Auburn area. The eNodeBs are also selected to each have a bandwidth of 10 MHz, corresponding to $M = 100$ and $\lambda_{\text{chip}} = 19.51$ m. This bandwidth is selected because it corresponds with the experimental data that is discussed later. In general, there are two main types of simulations performed. In the first, the receiver position is held constant and is termed 'static'. In the second, the receiver position and velocity vary according to a nearly constant velocity model, where acceleration is modeled by White noise with variance $\sigma_{x,y,z} = 2$ m^2/s^3 in all axes. The second type of simulation is termed 'dynamic' in the sequel. The static simulations will use

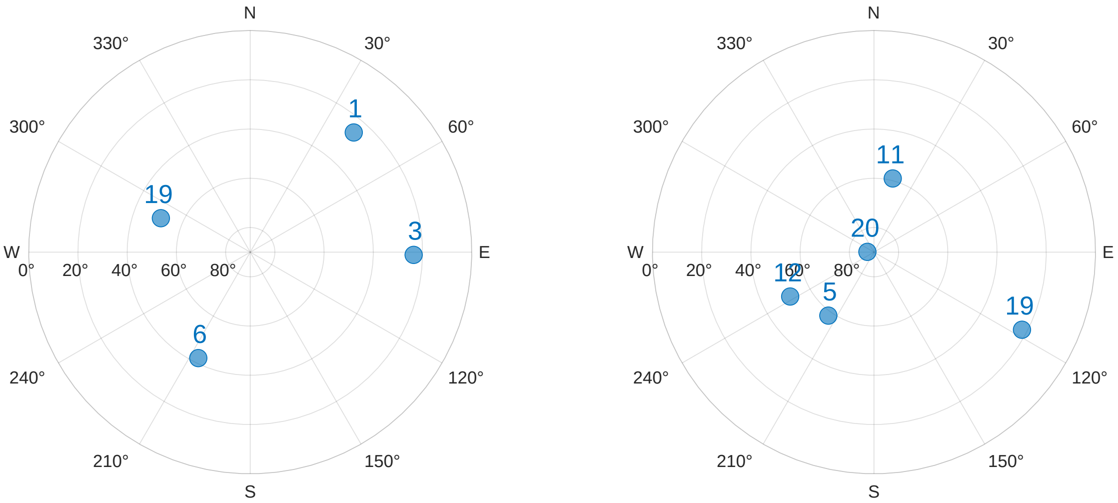

Figure 5.2: Static (left) and Dynamic (right) Skyplots.

the GPS satellites shown in the skyplot on the left in Figure 5.2; the dynamic simulations will use the GPS satellites shown in the skyplot on the right in Figure 5.2 because those satellites were observed during the dynamic experimental runs shown in Section 5.2. Each simulation will either be GPS-only, GPS and 1 eNodeB, GPS and 4 eNodeBs, or LTE-only. In the GPS and 1 eNodeB scenario, 'eNodeB 1' from Figure 5.1 is used. The Position Dilution of Precisions (PDOPs) for each scenario are shown in Table 5.2. It is shown that by combining GPS and LTE emitters, the positioning geometry improves dramatically. In the static and dynamic simulations, each GPS channel is subjected to an AWGN channel.

Table 5.1: Assigned center frequencies of each eNodeB.

eNodeB	**Center Frequency (MHz)**
1	885
2	751
3	885
4	2120

Table 5.2: The PDOPs of the possible configurations for each scenario.

Scenario	**GPS**	**GPS and 1 eNodeB**	**GPS and 4 eNodeB**
Static	22.41	5.13	1.57
Dynamic	4.97	4.20	1.24

The LTE channels are subjected to either an AWGN or multipath fading channel, depending on the simulation. In addition, the receiver's knowledge of the eNodeBs' positions is varied as well. To summarize, Table 5.3 shows the disturbances applied to the LTE signals. Note that the order does not matter and that any eNodeB position error can be coupled with any multipath delay profile, allowing for 16 different combinations. The multipath models shown in Table 5.3 are further described in Table 5.4. The multipath models increase in severity as Table 5.4 descends. 3GPP prescribes them to emulate realistic multipath conditions in pedestrian (EPA), vehicular (EVA), and urban (ETU) scenarios [69].

Table 5.3: eNodeB position estimation errors and multipath delay profiles [69] used for the simulations.

1-σ eNodeB Position RMSE (m)	**Multipath Delay Profile**
0	None
0.01	EPA
0.1	EVA
1	ETU

Table 5.4: Multipath channel models [69] used in the presented simulations.

Channel Model	**Channel Length**	**Path Delays (m)**	**Path Averaged Powers (dB)**
None	1	0	0.0
EPA	7	0, 9, 21, 27, 33, 57, 123	0.0, -1.0, -2.0, -3.0, -8.0, -17.2, -20.8
EVA	9	0, 9, 45, 93, 111, 213, 327, 519, 753	0.0, -1.5, -1.4, -3.6, -0.6, -9.1, -7.0, -12.0, -16.9
ETU	9	0, 15, 36, 60, 69, 150, 480, 690, 1500	-1.0, -1.0, -1.0, 0.0, 0.0, 0.0, -3.0, -5.0, -7.0

Finally, simulations termed 'outage' will be performed. These simulations are the same as the dynamic simulations, but the C/N_0 of all emitters will be held constant. For these simulations, the C/N_0 is held at 34 dB-Hz. This value is selected based on experimental data. Over the course of the simulations, the GPS signals will experience outages. During these outages, the GPS signals will all be instantaneously degraded to a C/N_0 of 10 dB-Hz for 2 seconds. This is to model a blockage caused by an overpass or tall building within a city. Since these things occur regularly as a vehicle traverses an urban environment, these outages will occur at intervals of 30 seconds. 100 Monte Carlo runs will be performed that each last 120 seconds. Scalar tracking, GPS-only VDFLL, and GPS and LTE VDFLLs will be compared. The main metric used is the survival rate of the GPS signals over the duration of the 120-second run.

5.1.2 Simulation Methodology

DARE

In the linear Kalman Filter (KF), the lower bound on performance $\mathbf{P}$ for a given state transition $\mathbf{F}$, process noise covariance $\mathbf{Q}$, observation matrix $\mathbf{H}$, and measurement covariance $\mathbf{R}$ is determined by using the standard KF equations given by

$$\begin{aligned} \mathbf{P}^{-} &= \mathbf{F}\mathbf{P}\mathbf{F}^{\mathrm{T}} + \mathbf{Q}^{\mathrm{T}}, \qquad (5.1) \\ \mathbf{K} &= \mathbf{P}^{-}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{P}^{-}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}, \\ \mathbf{P} &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^{-}(\mathbf{I} - \mathbf{K}\mathbf{H})^{\mathrm{T}} + \mathbf{K}\mathbf{R}\mathbf{K}^{\mathrm{T}}. \end{aligned}$$

In the EKF, these equations must be iterated until $\mathbf{P}$ converges to $\mathbf{P}_{\mathrm{SS}}$. The diagonal of $\mathbf{P}_{\mathrm{SS}}$ then describes the lower bound of the estimator's performance. At times, one may wish to know how well the measurements themselves are being tracked by the estimator. This is of particular interest with regard to signal tracking because it gives some insight into how well the estimator is tracking the symbol timing (pseudoranges) and the carrier frequency (pseudorange rates). To extract this information, the steady state covariance matrix $\mathbf{P}_{\mathrm{SS}}$ may be transformed into the measurement domain. This is done by using the observation matrix as

$$\mathrm{E}[\epsilon\epsilon^{\mathrm{T}}] = \mathbf{H}\mathbf{P}_{\mathrm{SS}}\mathbf{H}^{\mathrm{T}}, \qquad (5.2)$$

where ϵ is the tracked error of interest.

In this thesis, this process is applied to the VDFLL described in Chapter 4. The steady state covariance matrix $\mathbf{P}_{\mathrm{SS}}$ describes the distribution of the receiver's position, velocity, and clock states. Equation (5.2) is then used to determine the errors in pseudorange and pseudorange rate tracking. This method provides a lower bound on the VDFLL signal tracking performance error due to thermal noise. However, it does not account for other sources of error, such as multipath or emitter localization errors. Since these errors are more difficult to include, Monte Carlo simulations will be used.

Monte Carlo Simulation

To capture the effects of errors due to sources other than AWGN Monte Carlo simulations are performed. Additionally, the probability of tracking the GPS and LTE signals can be observed. For each simulation in this thesis, 100 Monte Carlo runs will be performed. Figure 5.3 shows a block diagram summary of the Monte Carlo simulations. First, the true emitter and receiver states are used to generate the true pseudoranges and pseudorange rates. Then, the most recent estimate of the receiver's position according to the VDFLL is used to produce estimates of the pseudoranges and pseudorange rates. They are differenced with the true pseudoranges and pseudorange rates to determine the true error. These errors are scaled by the code or carrier wavelength. This results in errors in code phase in chips, and errors in carrier frequency in Hz. These values can be substituted into the correlator model presented in equation (3.4). Note that the autocorrelation function $R(\delta\tau)$ depends on the broadcasting signal. The autocorrelation functions for GPS and the SSS are given by

$$\begin{aligned} R_{\text{GPS}}(\delta\tau) &= \begin{cases} 1 - |\delta\tau|, & \text{if } |\delta\tau| < 1, \\ 0, & \text{if } |\delta\tau| \geq 1, \end{cases} \\ R_{\text{SSS}}(\delta\tau) &= \operatorname{sinc}(\delta\tau), \end{aligned} \tag{5.3}$$

where $\delta\tau$ is the code phase error in chips.

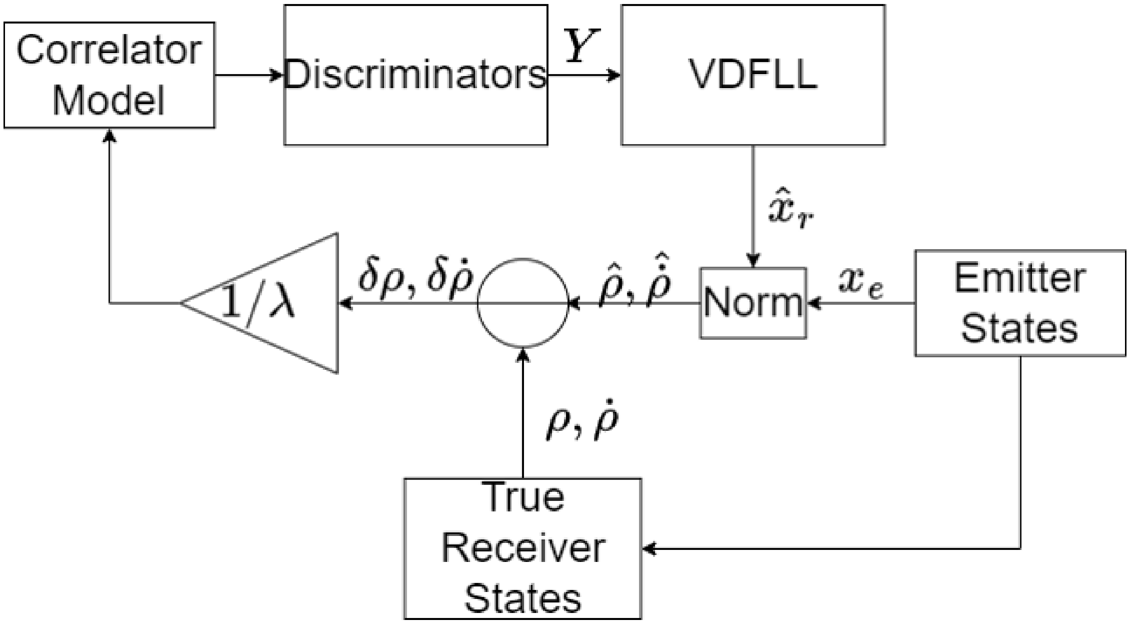

Figure 5.3: Monte Carlo simulation block diagram.

With regard to the CRS, a different correlator model must be used. Its prompt correlators are given by [39]

$$\begin{aligned} IP &= A\frac{\sin(\pi(\delta\tau))}{\sin(\pi(\delta\tau)/M)}\cos(-\pi(\delta\tau)(M-1/M)-(2\pi f_{\text{err}}T_{\text{symbol}}+\theta_{\text{err}}))+\eta_I, \\ QP &= A\frac{\sin(\pi(\delta\tau))}{\sin(\pi(\delta\tau)/M)}\sin(-\pi(\delta\tau)(M-1/M)-(2\pi f_{\text{err}}T_{\text{symbol}}+\theta_{\text{err}}))+\eta_Q. \end{aligned} \tag{5.4}$$

The early and late correlators can be extrapolated easily, so they will be omitted here. The discriminators used in the simulations, and then later in the experiments, are the same that were presented in Chapter 3. In Figure 5.3 the VDFLL block possesses a local estimate of the emitters' states. This estimate always stays perfect for GPS but will change for LTE, depending on the simulation (Table 5.3). That is, the VDFLL will be constructing the observation matrix **H** using its own knowledge of the emitter states not necessarily the true emitter states. Finally, to apply multipath to the LTE signals, two main methods will be used. For the SSS, the multipath is applied to the correlators themselves. Once the path delays and powers are generated for the desired delay profile, a separate correlator is generated for each path delay and power. Then, the resultant correlators are summed. For example, an in-phase prompt correlator affected by multipath would be written as

$$IP = A\sum_{l=0}^{L-1} IP_l, \tag{5.5}$$

where $IP_l = \text{Re}(\alpha_l)R(\delta\tau - \tau_l)\text{sinc}(\pi f_{\text{err}}T_{\text{PDI}})\cos(\pi f_{\text{err}}T_{\text{PDI}} + \delta\phi + \phi_l)$, α_l is the complex power of the l-th tap, τ_l is the l-th tap delay, ϕ_l is the l-th tap phase advance, and L is the number of multipath taps.

For the CRS, the multipath effect is added to the discriminator outputs themselves. The code phase discriminator can be rewritten as [39]

$$\begin{aligned}\delta\tau &= |\mathcal{R}_e|^2 - |\mathcal{R}_l|^2 \\ &= |h_0|^2 M^2 A^2 S(\epsilon, \Delta) + \chi_1 + \chi_2,\end{aligned}$$

where h_0 is the gain and delay of the LOS path, M is the number of subcarriers containing the CRS, $S(\epsilon, \Delta)$ is the discriminator S-curve, and ϵ is the true symbol timing error. Then [39],

$$\begin{aligned}\chi_1 = {}& A^2 \left| \sum_{m=0}^{M-1}\sum_{l=1}^{L-1} h_l e^{-j2\pi(m/M)((\tau_l/T)+\epsilon+\Delta)} \right|^2 \\ &- A^2 \left| \sum_{m=0}^{M-1}\sum_{l=1}^{L-1} h_l e^{-j2\pi(m/M)((\tau_l/T)+\epsilon-\Delta)} \right|^2 \\ \chi_2 = {}& 2A^2 Re\left[\left(\sum_{m=0}^{M-1} h_0 e^{-j(2\pi/M)m(\epsilon+\Delta)} \right) \times \left(\sum_{m'=0}^{M-1}\sum_{l=1}^{L-1} h_l^* e^{j2\pi(m'/M)((\tau_l/T)+\epsilon+\Delta)} \right) \right] \\ &- 2A^2 Re\left[\left(\sum_{m=0}^{M-1} h_0 e^{-j(2\pi/M)m(\epsilon-\Delta)} \right) \times \left(\sum_{m'=0}^{M-1}\sum_{l=1}^{L-1} h_l^* e^{j2\pi(m'/M)((\tau_l/T)+\epsilon-\Delta)} \right) \right].\end{aligned}$$

Of course, multipath does not only affect the code phase. The carrier phase discriminator would be affected as well, and as a result, the frequency discriminator would be affected, too. The carrier phase discriminator can be rewritten as [22]

$$\delta\phi = \arg\left[\mathcal{R}_p + \chi_{\mathrm{PLL}}\right],$$

where $\mathcal{R}_p = IP + 1jQP$ and [22]

$$\chi_{\mathrm{PLL}} = \sum_{m=0}^{M-1}\sum_{l=1}^{L-1}\sqrt{C}e^{j\Delta\phi}\alpha_l e^{-j2\pi(m/M)(\tau_l/T)}. \tag{5.6}$$

5.1.3 Simulation Results

In this section, simulation results using the above setup are presented. The signal-tracking performance of the combined VDFLL is shown. It is compared with the traditional GPS-only VDFLL and with scalar tracking. The GPS-only VDFLL is tuned in the same way as the combined GPS/LTE VDFLL. That is, $\sigma^2_{x,y,z} = 0$ m^2/s^3 in the static case and $\sigma^2_{x,y,z} = 2$ m^2/s^3 in the dynamic case. The scalar tracking bandwidths remain the same for both cases and are purely estimates derived from loop theory due to thermal noise. These estimates assume tracking with an FLL and DLL. The bandwidths for GPS are set to $B_{\mathrm{FLL}} = 2$ Hz and $B_{\mathrm{DLL}} = 1$ Hz. The bandwidths for LTE are set to $B_{\mathrm{FLL}} = 0.2$ Hz and $B_{\mathrm{DLL}} = 0.01$ Hz. In addition to the estimates produced by loop theory, 100-run Monte Carlo simulations are performed for scalar tracking of the LTE signals to analyze the probability of tracking and the effect of multipath. The Monte Carlo simulations are also performed with an FLL and DLL at the same bandwidths for comparison.

AWGN Simulations

GPS and 1 eNodeB Simulations: In the first set of simulations, the signals will only be affected by AWGN. The receiver will have varying knowledge about the location of the LTE emitters, however. Figure 5.4 shows the results of the static simulation with 4 GPS satellites and 1 eNodeB. The LTE signals are tracked using the SSS with varying knowledge of the emitter position. Figure 5.4 shows that by including the 1 eNodeB tracking the SSS, the GPS tracking performance improves only marginally when compared with GPS-only vector tracking. It is also shown that the code phase tracking of the SSS is negatively affected by the lack of knowledge of the eNodeB position. It should be noted that this error does not negatively affect the code tracking of the GPS signals. The carrier tracking of both signals is not affected by this error either. Additionally, it is shown that the theoretical results are optimistic at low C/N_0. This is expected because the DARE does not account for the loss of lock. The DARE approach assumes that we can track at any C/N_0. However, it is still useful because it can be used to predict when a loss of lock may occur. Often this occurs when a certain 1-σ threshold is crossed. Some examples of these thresholds are shown in Figure 5.4. These thresholds indicate when the discriminator function leaves the linear region, where it is trackable. For more information on these thresholds see [45]. The same 1-σ thresholds are used for LTE as are GPS (i.e., 1/6 of a chip).

Figure 5.5 shows the probability of tracking both signals during the static simulations. The probability of tracking is computed as the number of signals that are tracked for the entirety of the simulation averaged over the total number of Monte Carlo simulations. It should be noted that the LTE signals' probability of tracking is compared with scalar tracking; whereas, the GPS signals' probability of tracking is compared with GPS-only vector tracking. It is shown that despite the minimal increase in tracking performance granted by including the SSS, the probability of tracking the GPS signals begins to lose lock later than the GPS-only VDFLL. The combined VDFLL maintains lock of the SSS much longer than scalar tracking as well.

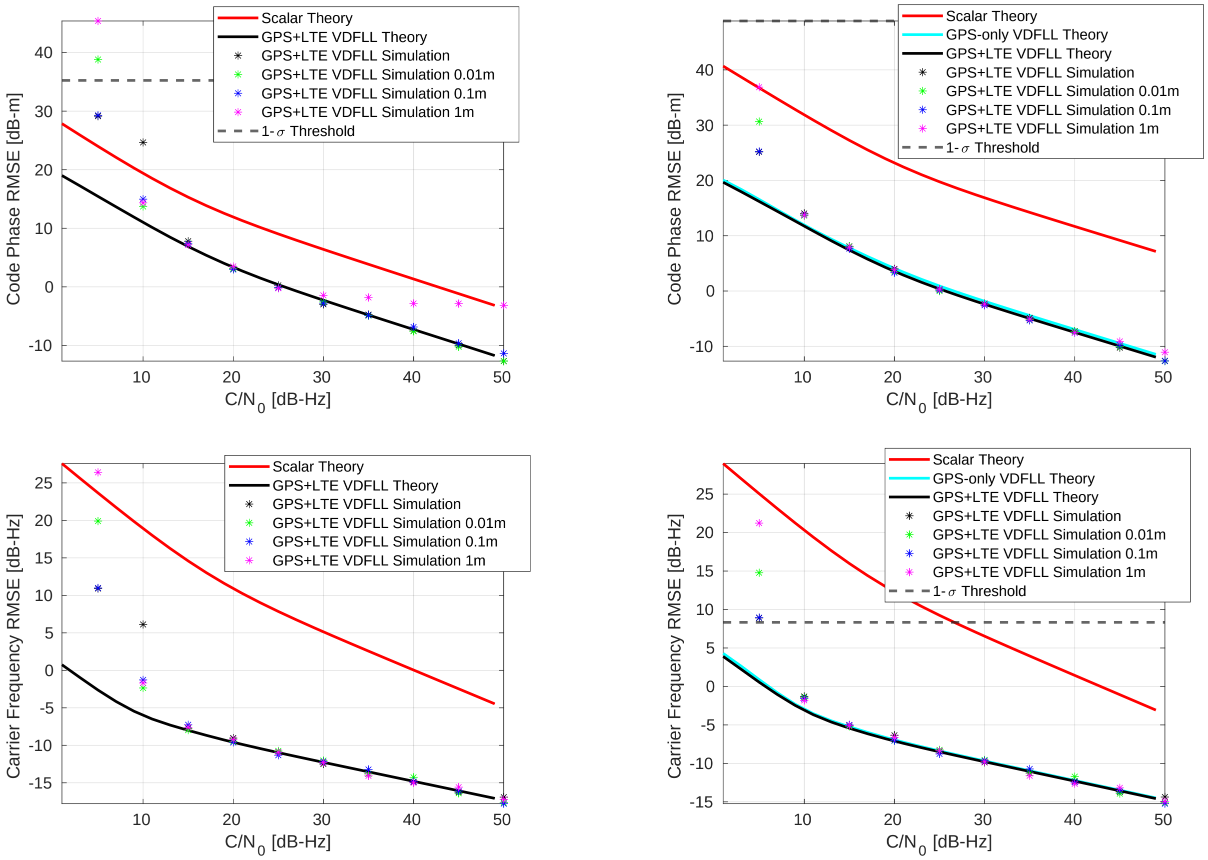

Figure 5.4: 4 GPS and 1 eNodeB static simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the SSS. The GPS tracking results are shown on the right.

Next, the 5 GPS and 1 eNodeB dynamic simulations are performed. Figure 5.6 shows the results when tracking the SSS. Again, the error in eNodeB localization only affects the code phase tracking of the LTE signal when it exceeds 1 m. This time, due to the dynamics, the GPS code phase tracking is degraded as well at high C/N_0. The GPS code tracking is still only marginally improved when compared with the GPS-only VDFLL. Both GPS and LTE code tracking are improved significantly when compared with scalar tracking. The carrier tracking, while worse than the static case, remains superior to scalar tracking. For the most part, it is unaffected by eNodeB localization errors as well. Figure 5.7 shows the probability of tracking the GPS and LTE signal in the aforementioned dynamic case, while tracking the SSS. It is shown that there is still significant tracking improvement of the LTE

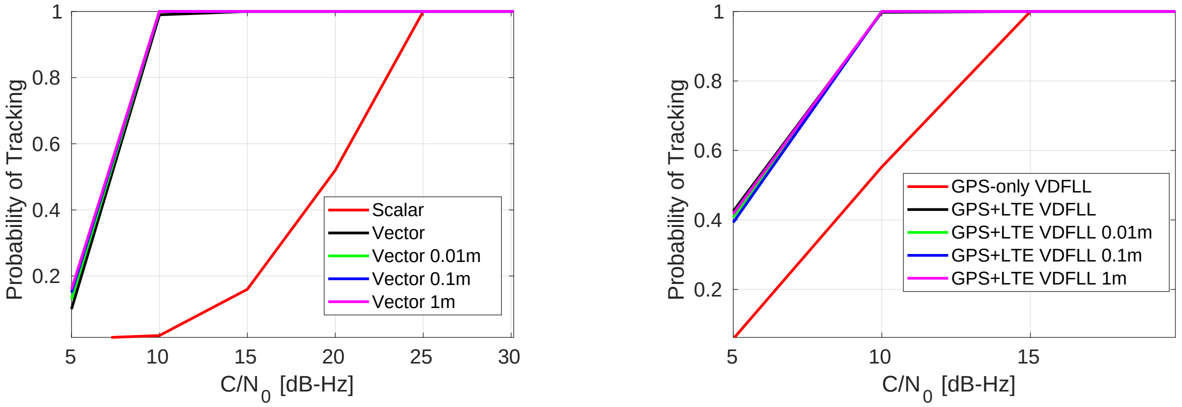

Figure 5.5: Probability of tracking LTE (left) and GPS (right) for the 4 GPS and 1 eNodeB static simulations. The LTE signals are tracked using the SSS.

signal when compared with scalar tracking. Additionally, the probability of tracking the GPS signal improves considerably when compared with GPS-only vector tracking.

The 4 GPS and 1 eNodeB static simulations are also performed with CRS tracking instead of the SSS. Figure 5.8 shows the results of these simulations. It is shown that the code phase tracking of the CRS is only marginally improved by the combined GPS/LTE VDFLL when compared with scalar tracking. This is caused by the tight scalar DLL bandwidth and the very low open-loop code phase variance incurred by the CRS discriminator. Nonetheless, there is some improvement. The 1-σ threshold should be noted as well. The CRS is much more likely to lose lock because of code phase error instead of frequency error. This is because the CRS must be tracked to sample-level precision. The width of the sample when tracking the CRS when $M = 100$ is just 19.51 m. Using the rule-of-thumb threshold provided in [45] and the S-curve shown in [39], the threshold is merely 2.17 m. The code phase error induced by the eNodeB localization error does not affect the CRS much in this case. Conversely, it affects GPS signal tracking significantly. This is due to the amount of trust that is placed on the CRS as a result of its low open-loop code phase variance. When the eNodeB localization error is less than 1 m, the code phase tracking of the GPS signal is improved more than with the SSS, but it is still a marginal improvement. The LTE carrier tracking is improved significantly when compared with scalar tracking. It is also not heavily affected

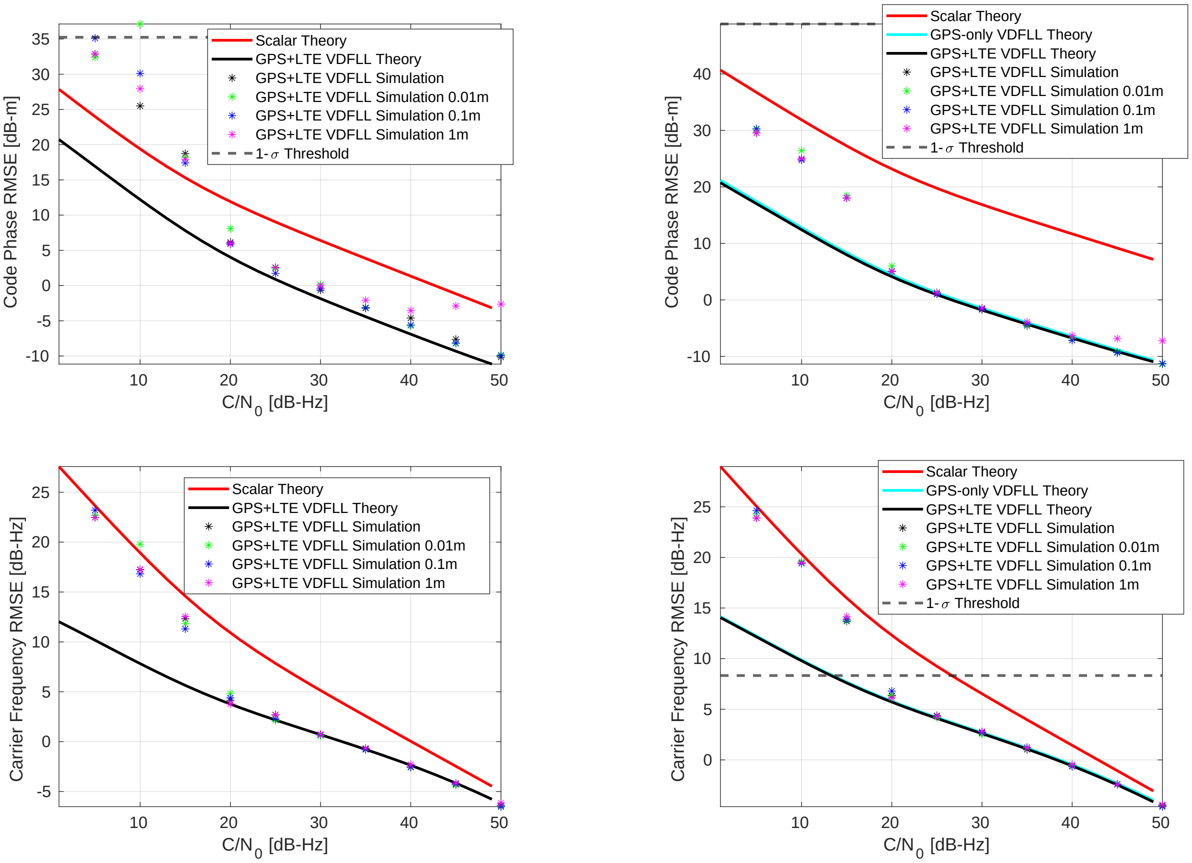

Figure 5.6: 5 GPS and 1 eNodeB dynamic simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the SSS. The GPS tracking results are shown on the right.

by eNodeB localization errors. The GPS carrier frequency tracking improves considerably when compared with scalar tracking. In addition, the carrier frequency tracking is improved much more than with the GPS-only VDFLL or when the SSS was used, instead. Figure 5.9 shows the probability of tracking the GPS and LTE signals in this scenario. Notably, the CRS loses lock more easily in vector tracking when compared with scalar tracking. The tracking begins to break down before the 1-σ threshold would indicate, but about where the threshold indicates for scalar tracking. The combined VDFLL still improves GPS signal tracking probability when compared with the GPS-only VDFLL, but because the LTE signal is lost earlier, it is not as large of an improvement as seen with the SSS.

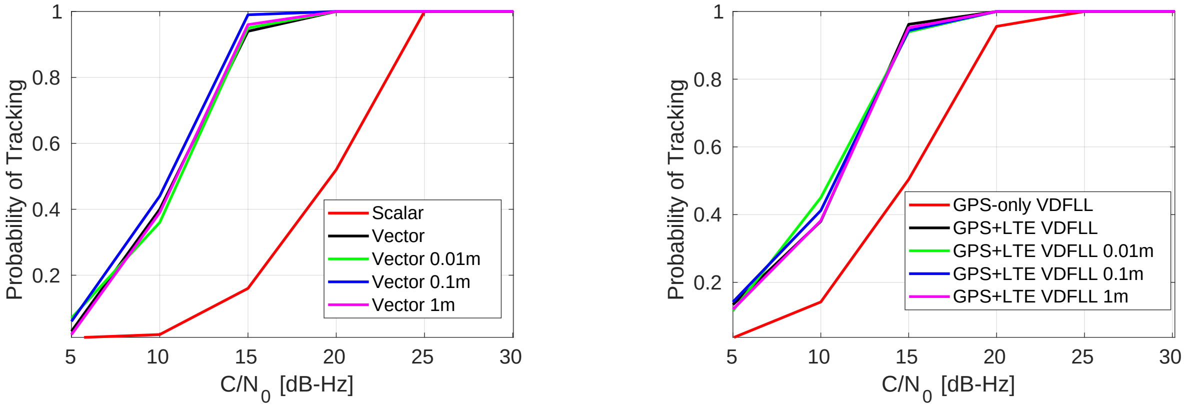

Figure 5.7: Probability of tracking LTE (left) and GPS (right) for the 5 GPS and 1 eNodeB dynamic simulations. The LTE signals are tracked using the SSS.

Figure 5.10 shows the results of the dynamic simulation with 5 GPS satellites and 1 eNodeB, where the LTE signals are tracked using the CRS. The results indicate that the code tracking does not change much between scalar and vector for the LTE signal. The code phase tracking is somewhat better at lower C/N_0 in the vector case and somewhat better at high C/N_0 in the scalar case. The effect of eNodeB localization error is minimal. However, the code phase tracking error for GPS is large when the eNodeB localization error is greater than 1 m. Otherwise, it outperforms both the scalar and GPS-only VDFLL tracking solutions. The carrier frequency tracking is mostly the same. The LTE signal is tracked better at lower C/N_0 and worse at higher C/N_0 in the combined VDFLL when compared with scalar tracking. The GPS carrier frequency tracking improves modestly when compared with scalar tracking and the GPS-only VDFLL. However, if the eNodeB localization error is 1 m the GPS carrier frequency tracking degrades as well. These results overall indicate that large errors in eNodeB localization tend to degrade the overall navigation solution when the CRS is tracked. Figure 5.11 shows the probability of tracking the LTE and GPS signals in this scenario. It is shown that scalar tracking still outperforms the combined VDFLL at tracking the LTE signal with the CRS. However, the probability of tracking the GPS signal is still improved when compared with the GPS-only VDFLL. [hbt]

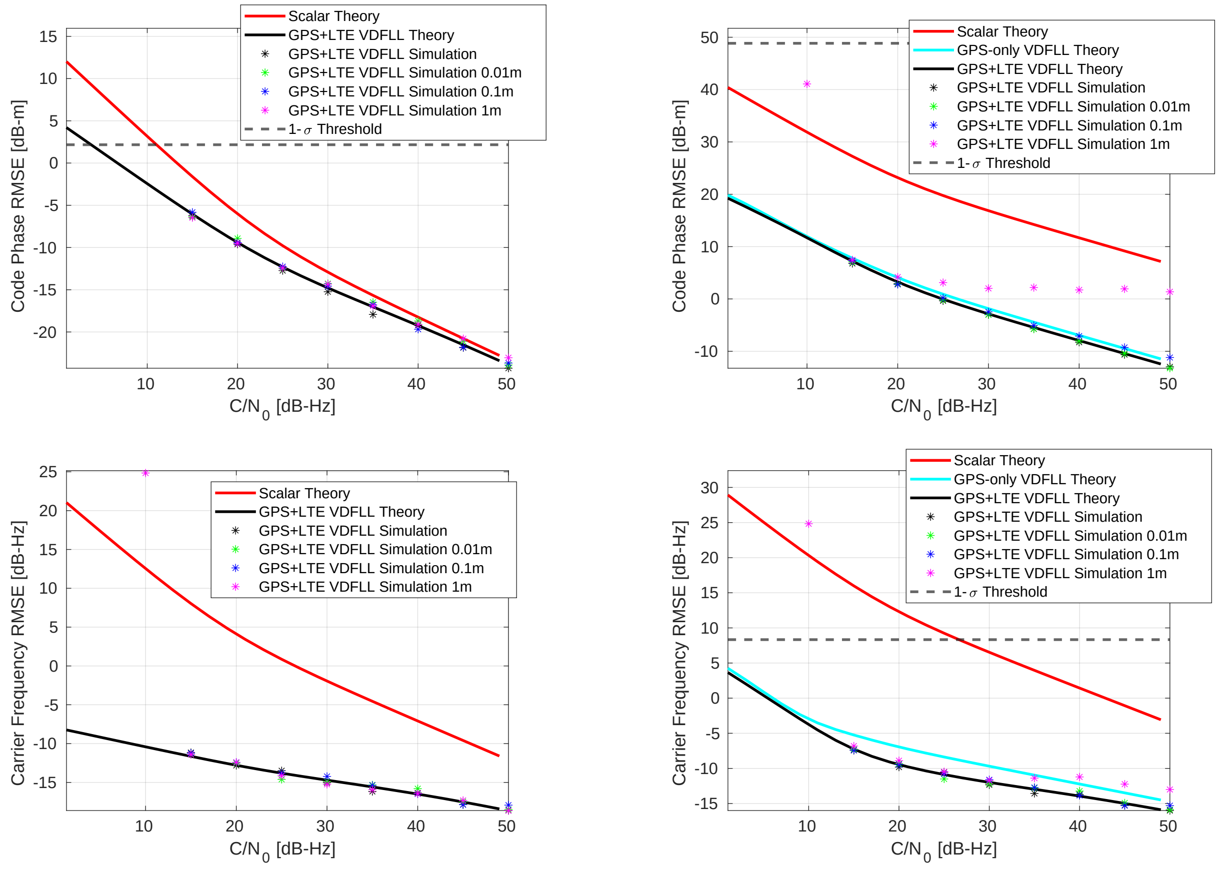

Figure 5.8: 4 GPS and 1 eNodeB static simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the CRS. The GPS tracking results are shown on the right.

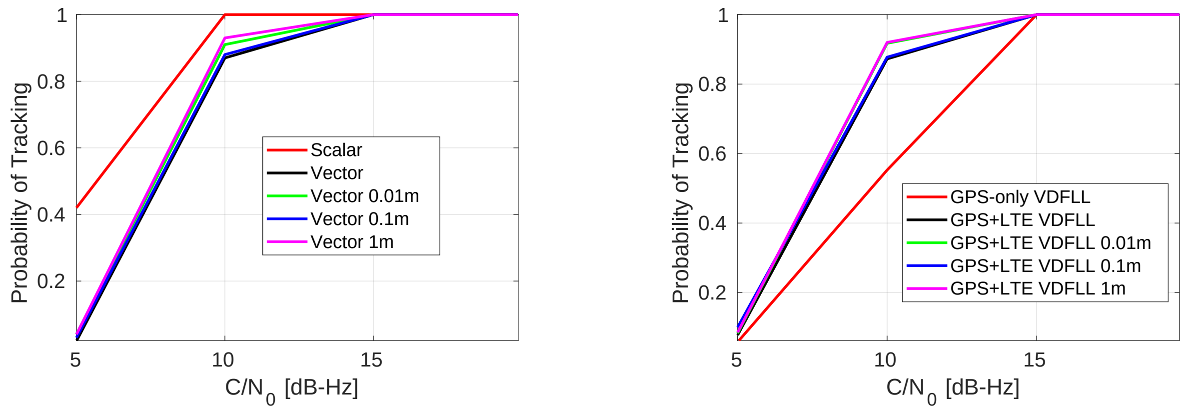

Figure 5.9: Probability of tracking LTE (left) and GPS (right) for the 4 GPS and 1 eNodeB static simulations. The LTE signals are tracked using the CRS.

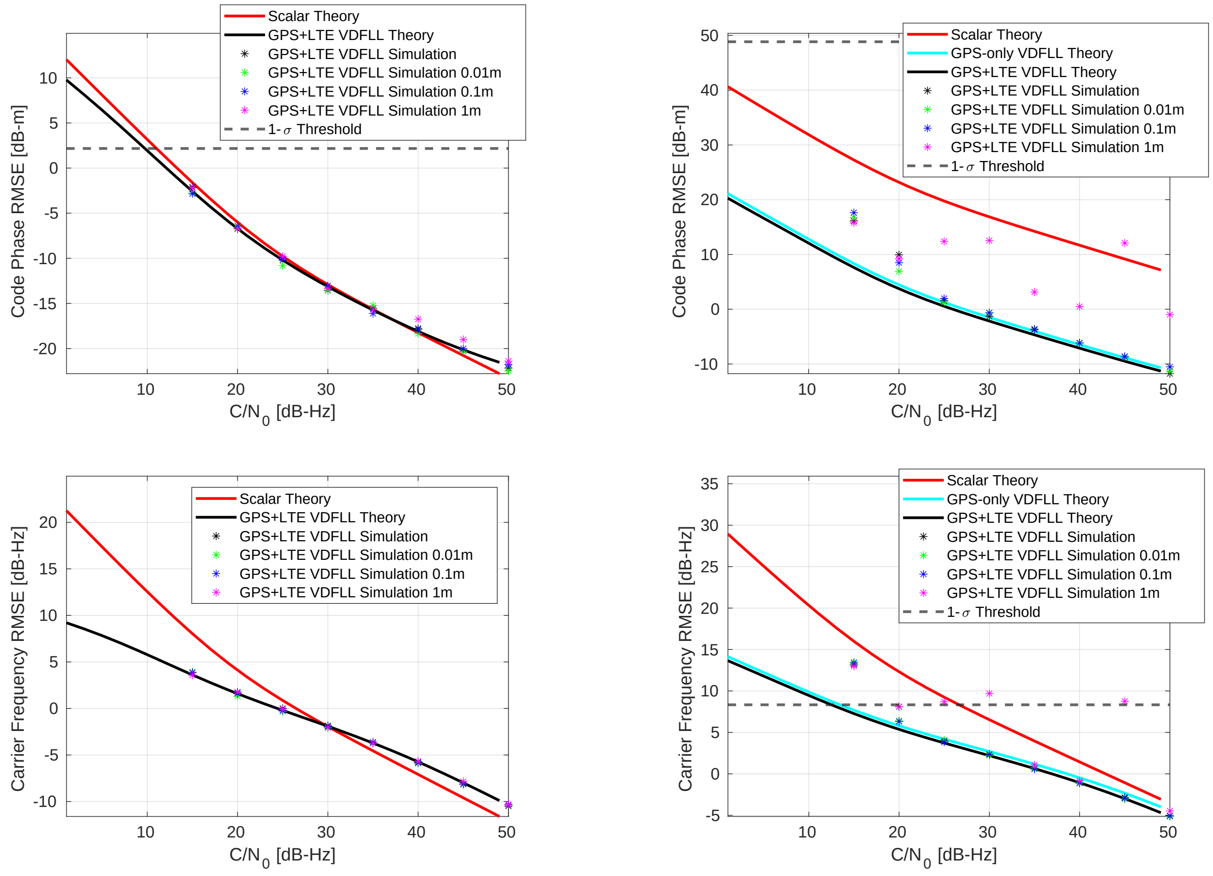

Figure 5.10: 5 GPS and 1 eNodeB dynamic simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the CRS. The GPS tracking results are shown on the right.

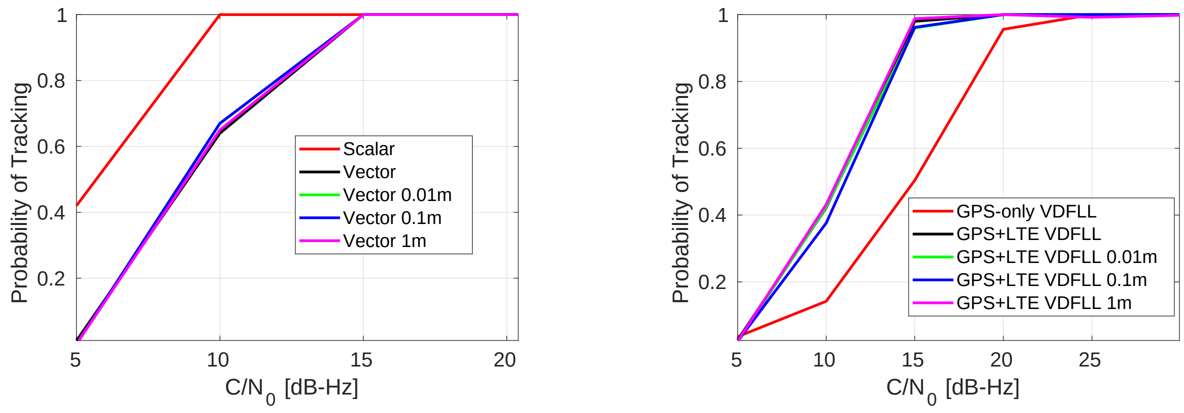

Figure 5.11: Probability of tracking LTE (left) and GPS (right) for the 5 GPS and 1 eNodeB dynamic simulations. The LTE signals are tracked using the CRS.

GPS and 4 eNodeB Simulations: In the second set of simulations the advantages and consequences of including 4 eNodeBs in the combined VDFLL are analyzed. Figure 5.12 shows the RMSE results of the static 4 GPS and 4 eNodeB simulations. It is shown that by including multiple eNodeBs all with similar localization errors the overall code phase tracking error is approximately the same when compared with the results in Figure 5.4, which is the same scenario with 1 eNodeB. However, if the receiver has perfect knowledge of the eNodeB positions there is mild improvement over what is shown in Figure 5.4. The largest improvement comes in the code phase tracking of the GPS signal, outperforming the GPS-only VDFLL modestly. The carrier frequency tracking of the LTE signal outperforms that of Figure 5.4 regardless of eNodeB localization error. Finally, the GPS carrier frequency tracking improves over the GPS-only VDFLL primarily at lower C/N_0. All tracking is improved significantly when compared with scalar tracking. Figure 5.13 shows the probability of tracking the LTE and GPS signals throughout the simulations. It is shown that regardless of eNodeB localization errors, the probability of tracking is improved significantly when compared with scalar tracking and the GPS-only VDFLL for the LTE and GPS signals, respectively.

Figure 5.14 shows the results of the 5 GPS and 4 eNodeB dynamic simulations where the LTE signal is tracked using the SSS. The code phase tracking of the LTE signal does not improve much when compared with Figure 5.6. The error incurred by the eNodeB localization error is the same for both LTE and GPS code phase tracking. However, GPS code phase tracking improves when compared with the GPS-only VDFLL. It also improves more when compared with what is shown in Figure 5.6, Indicating that overall positioning performance should improve. However, the degradation that may occur as a result of the knowledge of the eNodeB location is unlikely to harm the solution more than the 1 eNodeB case. The carrier frequency tracking of the LTE signal is comparable with that shown in Figure 5.6. The GPS carrier frequency tracking improves modestly over the GPS-only VDFLL and when compared with Figure 5.6. All results improve significantly over traditional

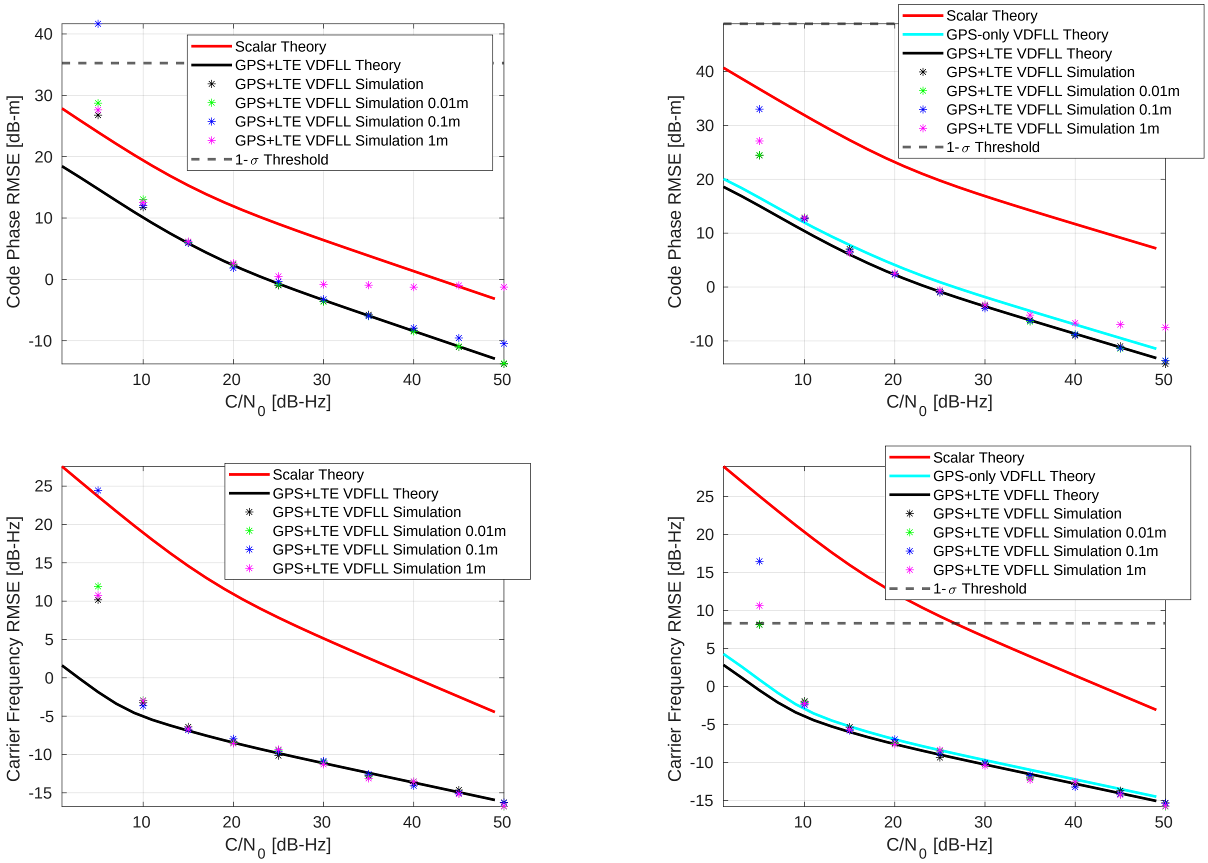

Figure 5.12: 4 GPS and 4 eNodeB static simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the SSS. The GPS tracking results are shown on the right.

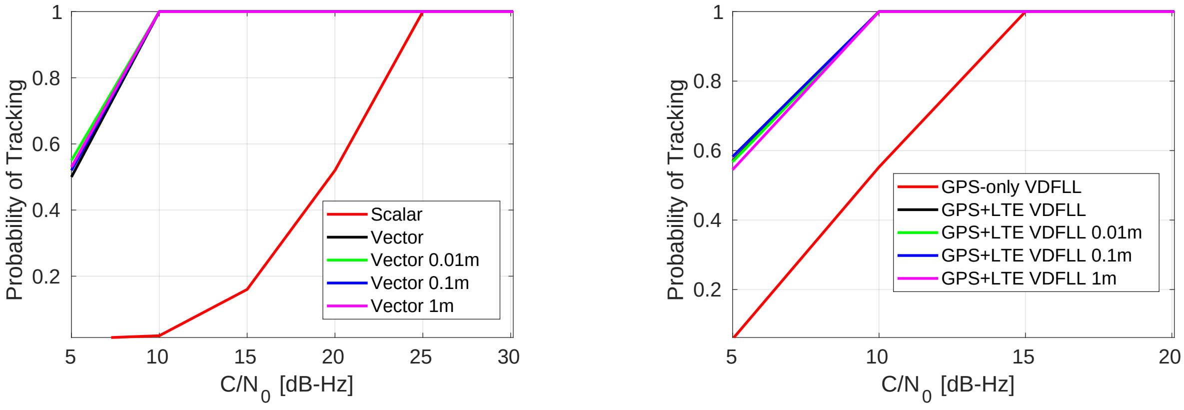

Figure 5.13: Probability of tracking LTE (left) and GPS (right) for the 4 GPS and 4 eNodeB static simulations. The LTE signals are tracked using the SSS.

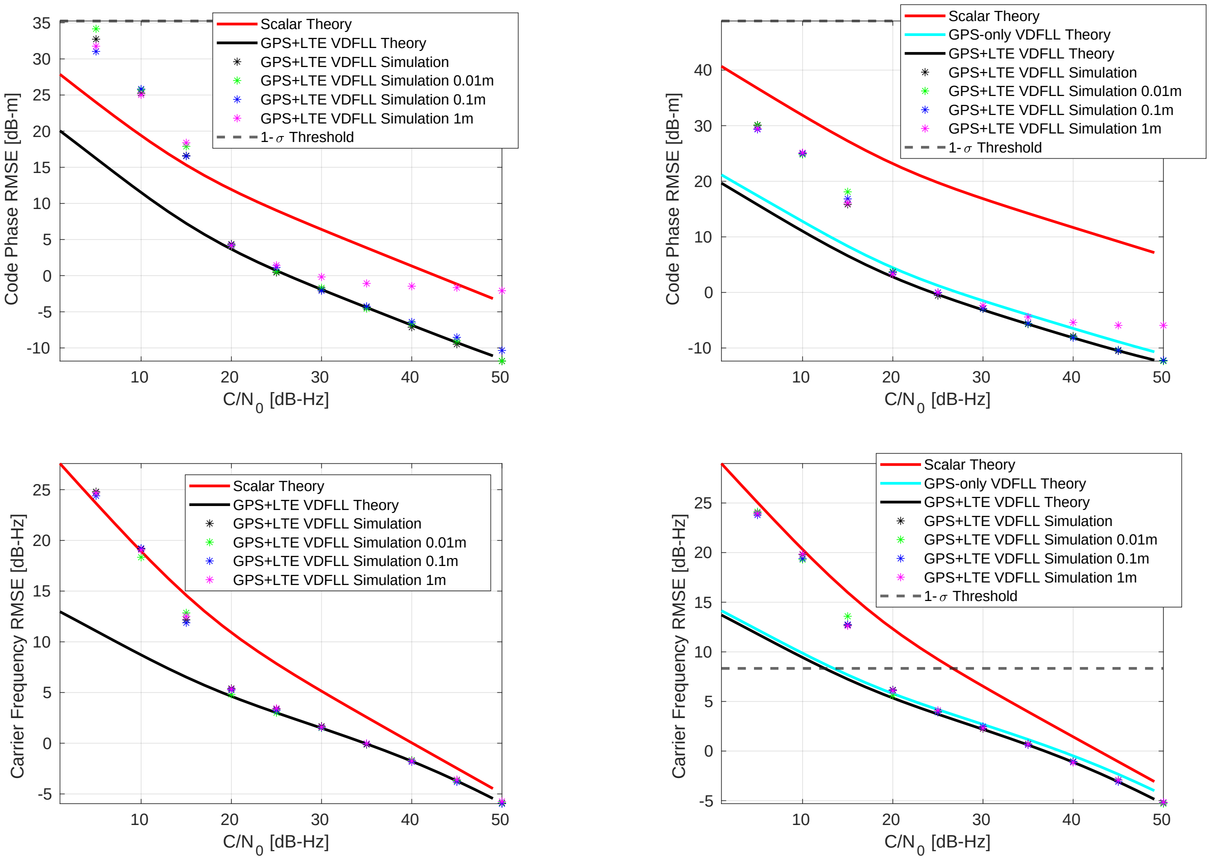

Figure 5.14: 5 GPS and 4 eNodeB dynamic simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the SSS. The GPS tracking results are shown on the right.

scalar tracking at low C/N_0. The carrier frequency tracking of the combined VDFLL is comparable to scalar tracking at high C/N_0. Figure 5.15 shows the probability of tracking the LTE and GPS signals in this scenario. It shows little to no improvement over the 1 eNodeB case shown in Figure 5.7.

Figure 5.16 shows the results of the 4 GPS and 4 eNodeB static simulations, where the LTE signal is tracked using the CRS. It is shown that the code phase tracking of the LTE signal is not improved much when compared with Figure 5.8. Conversely, the eNodeB localization error becomes a dominant source of error in the LTE code phase tracking. One could see that, with enough localization errors, the LTE signal could lose track at any C/N_0. In fact, it would not take much more than 1 m of error to cause this. In addition, this

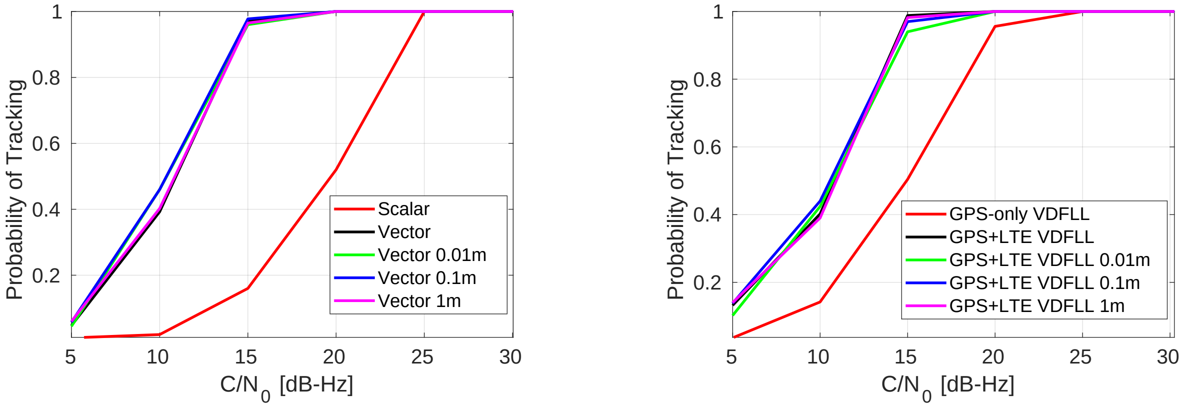

Figure 5.15: Probability of tracking LTE (left) and GPS (right) for the 5 GPS and 4 eNodeB dynamic simulations. The LTE signals are tracked using the SSS.

error now begins to cause significant errors in the GPS code and carrier tracking as well. At even 0.1m of error, it is expected that the position and velocity solution would be degraded. On the other hand, if the eNodeB location is known with high accuracy, the GPS code phase tracking is improved significantly when compared with the GPS-only VDFLL and with the results presented in Figure 5.8. The LTE carrier tracking is not improved much when compared with Figure 5.8. However, the GPS carrier frequency tracking improves greatly when compared with the GPS-only VDFLL, and the results are shown in Figure 5.8. Yet, it is also degraded by the eNodeB localization error. All results offer significant improvement over scalar tracking. Figure 5.17 shows the probability of tracking the LTE and GPS signal throughout these simulations. These results improve slightly when compared with Figure 5.9. Both the LTE and GPS signals are tracked 100% of the time at 10 dB-Hz. However, the LTE signal still underperforms when compared with scalar tracking.

Figure 5.18 shows the results of the 5 GPS and 4 eNodeB dynamic simulations, where the LTE signal is tracked using the CRS. The results indicate that the LTE code phase tracking is affected much in the same way as the static results shown in Figure 5.16. The main difference is that the code phase tracking error is much more comparable to the scalar tracking code phase error. At high C/N_0 the scalar tracking code phase error is shown to be less than the combined VDFLL. Next, the eNodeB localization error affects the GPS code

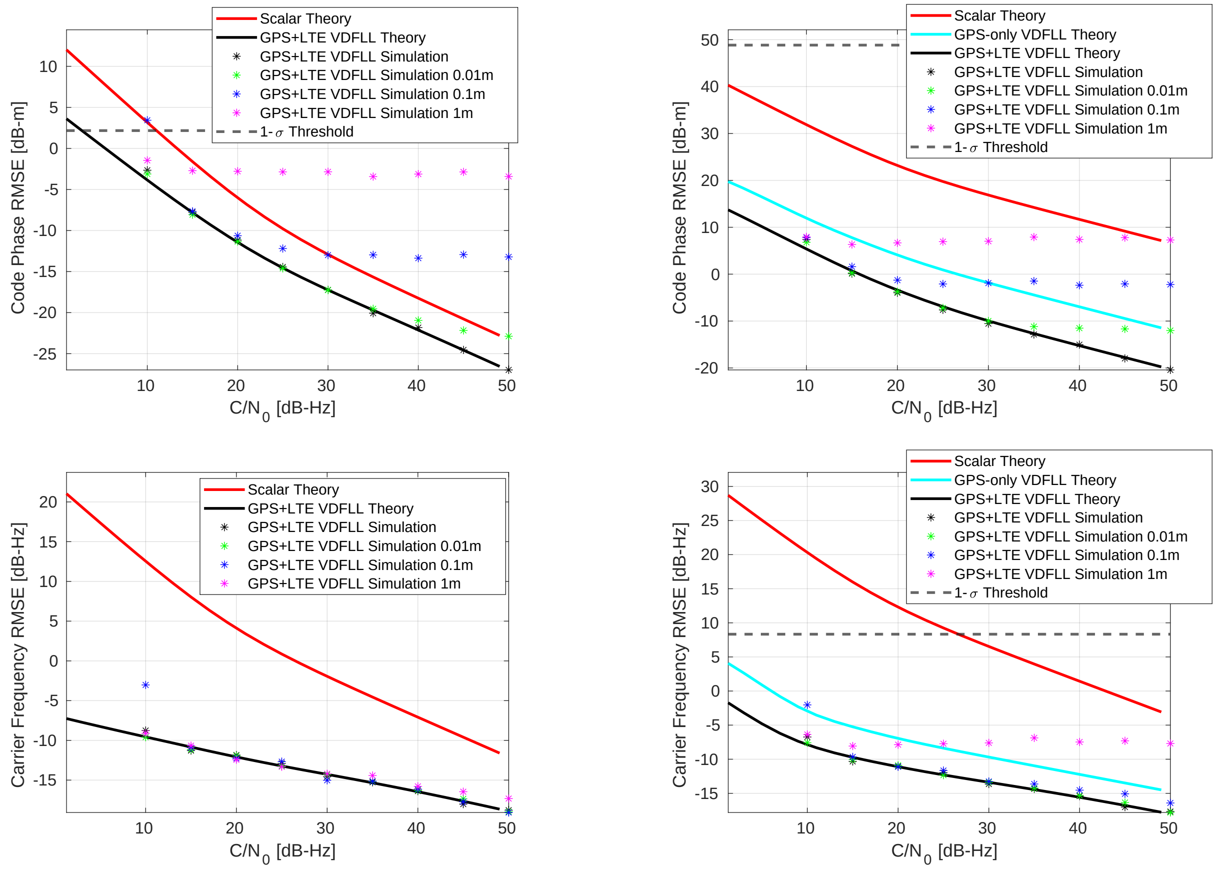

Figure 5.16: 4 GPS and 4 eNodeB static simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the CRS. The GPS tracking results are shown on the right.

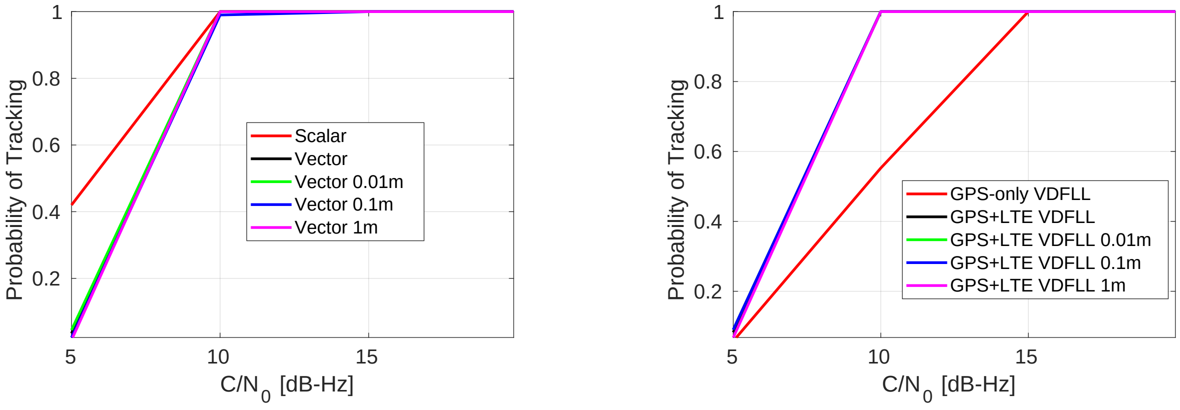

Figure 5.17: Probability of tracking LTE (left) and GPS (right) for the 4 GPS and 4 eNodeB static simulations. The LTE signals are tracked using the CRS.

phase tracking error in the same way as in Figure 5.16, and consistently worse than in Figure 5.10. Yet, in the best case, the code phase tracking error improves when compared with the GPS-only VDFLL and the results in Figure 5.10. The LTE carrier frequency tracking is more affected by the eNodeB localization error than in Figure 5.10, which leads to a more consistent error in the GPS carrier frequency tracking as well. However, the GPS carrier frequency tracking still improves considerably in the best case. In the best case, except at high C/N_0, the combined VDFLL outperforms the scalar tracking for LTE. In the best case, the combined VDFLL outperforms GPS scalar tracking at all C/N_0. Figure 5.19 shows the probability of tracking the LTE and GPS signals throughout the simulations. The combined VDFLL still loses lock of the LTE signal earlier than scalar tracking. The GPS signal tracking probability is still improved when compared with the GPS-only VDFLL. However, by including multiple eNodeBs the the probability of tracking both LTE and GPS increases mildly when compared with the results in Figure 5.11, where only 1 eNodeB is tracked.

In addition to the results presented above, the same simulations were performed when no GPS satellites were present. They are given in Appendix A and the figures therein. The largest change was in the probability of tracking the LTE signals, which improves slightly when GPS signals are also present.

Value of DARE

For characterizing the effects of eNodeB localization error and multipath, Monte Carlo simulation is necessary, but it is cumbersome to perform. For evaluating the performance of tracking under normal conditions the DARE solution matches the Monte Carlo simulation almost perfectly. The DARE solution takes a fraction of the time to compute. In this case, the DARE solution is sufficient with the caveat that at low C/N_0 it will be optimistic. At low C/N_0 the 1-σ threshold should be considered to determine loss of lock and when theory will not match experimentation.

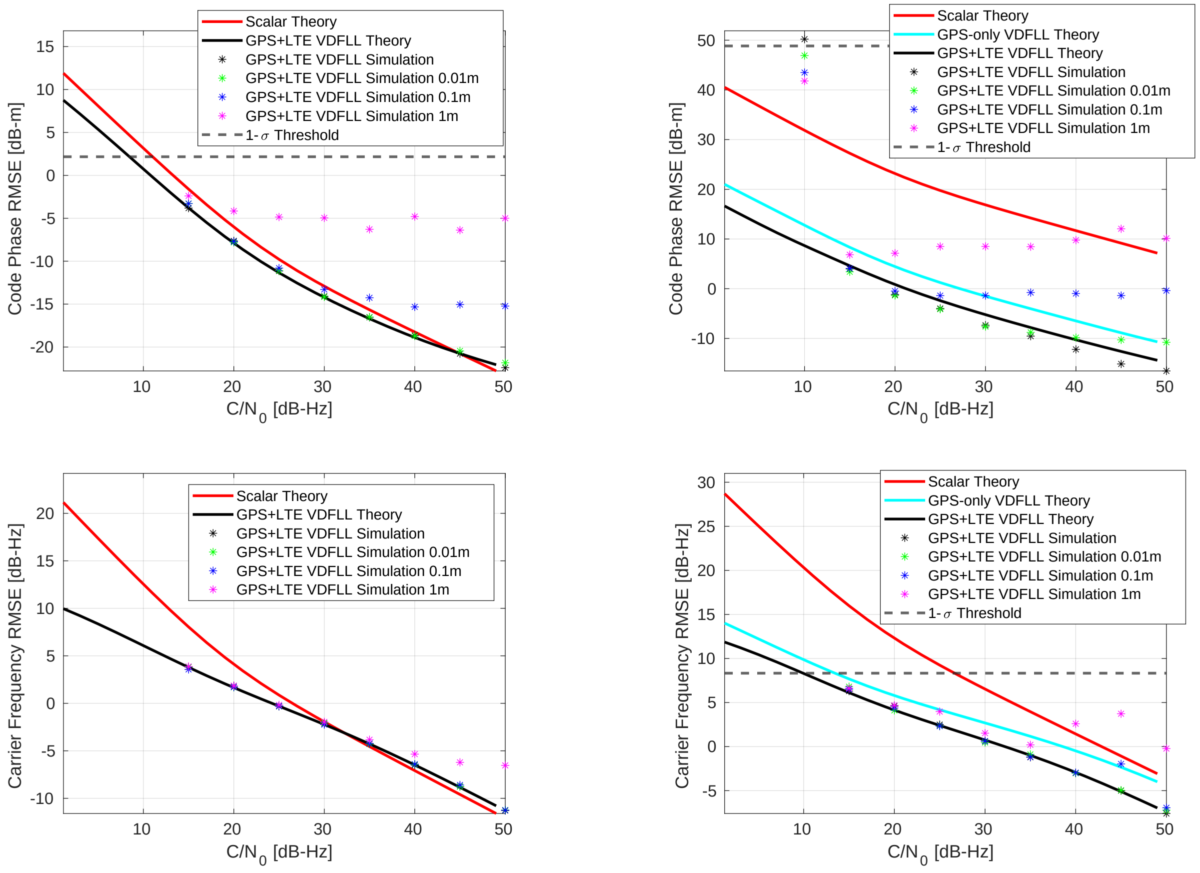

Figure 5.18: 5 GPS and 4 eNodeB dynamic simulation results. The LTE tracking results are shown on the left. The LTE signals are tracked using the CRS. The GPS tracking results are shown on the right.

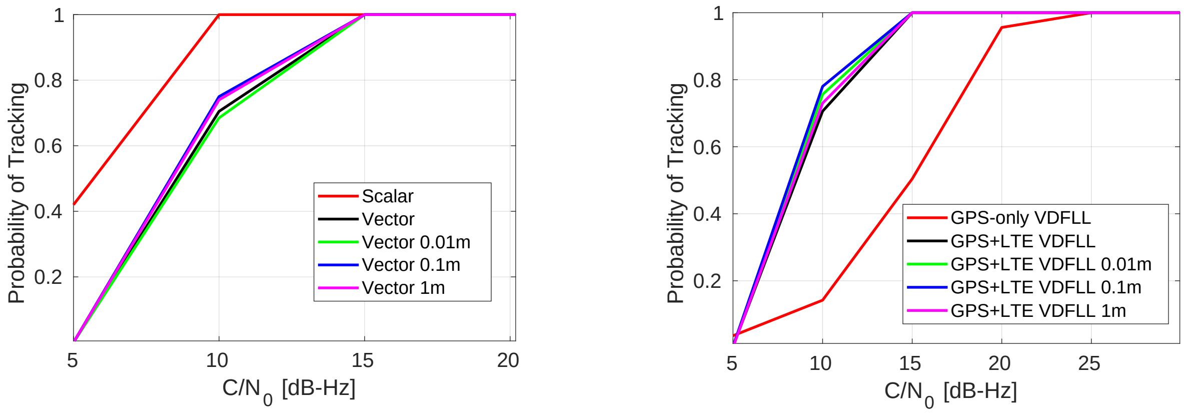

Figure 5.19: Probability of tracking LTE (left) and GPS (right) for the 5 GPS and 4 eNodeB dynamic simulations. The LTE signals are tracked using the CRS.

Multipath Simulations

Next, the same simulations are performed, but the LTE signals are subjected to the multipath delay profiles presented in Table 5.4. The code phase tracking of the LTE signals when the SSS and the CRS is analyzed. The probability of tracking the LTE signals is also evaluated. The multipath results are compared with the appropriate AWGN results. The combined VDFLL is then compared with scalar tracking subjected to the same multipath delay profiles.

GPS and 1 eNodeB Simulations: Figure 5.20 shows the code phase tracking results of the static and dynamic simulations, where the LTE signal is tracked using the SSS. It is shown that the presence of multipath causes a large amount of error when tracking the SSS. In the static case, the combined VDFLL tends to decrease the error when compared with scalar tracking. In the dynamic case, the results are somewhat improved at low C/N_0 but are compromised at high C/N_0 when the LTE signals are subjected to harsh multipath. The consequences of which are shown in Figure 5.21. The LTE signals are completely lost at high C/N_0 when subjected to the ETU multipath delay profile. When looking at Figure 5.20, it is likely that at high C/N_0 the same loss of lock would occur with the EVA multipath delay profile as well.

Figure 5.22 shows the results of the same simulations, where the LTE signals are tracked using the CRS. It should be noted that the overall signal tracking error is much less significant than with the SSS in Figure 5.20. Yet, vector tracking does not provide any improvement when compared with scalar tracking. There is not much difference between the static and dynamic simulations. Figure 5.23 shows the probability of tracking the LTE signal throughout the simulations. It is shown that the multipath channels, except ETU, do not affect the tracking probability when compared with AWGN. In the ETU channel, the LTE signal is lost much earlier than the rest. This is likely a result of the navigation filter overvaluing the CRS measurements. This will cause a large navigation solution error which will cause loss of

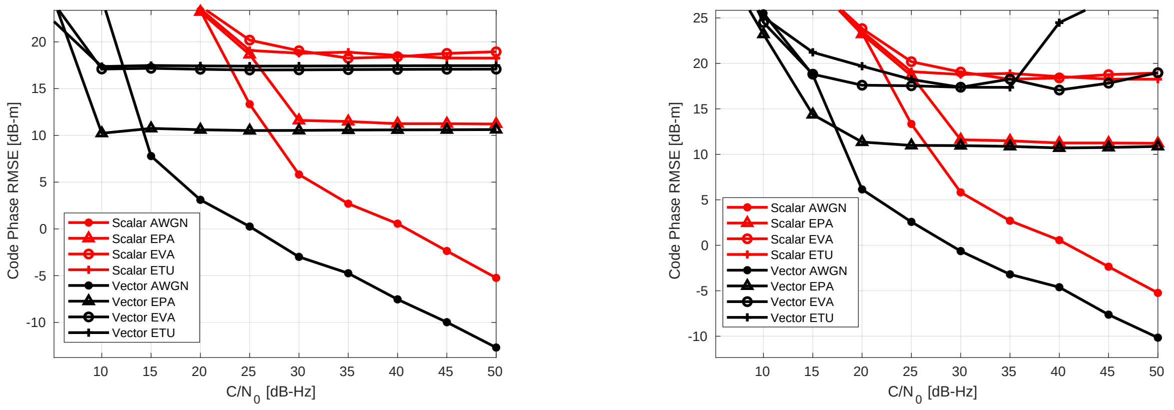

Figure 5.20: The code phase tracking performance of the 4 GPS and 1 eNodeB static simulations (left) and 5 GPS and 1 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS

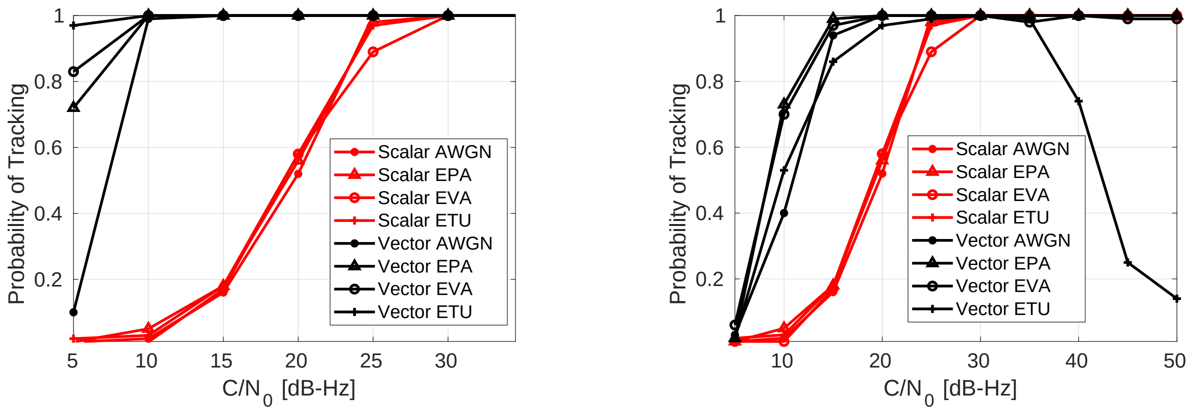

Figure 5.21: The probability of tracking the LTE signal during the 4 GPS and 1 eNodeB static simulations (left) and 5 GPS and 1 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS.

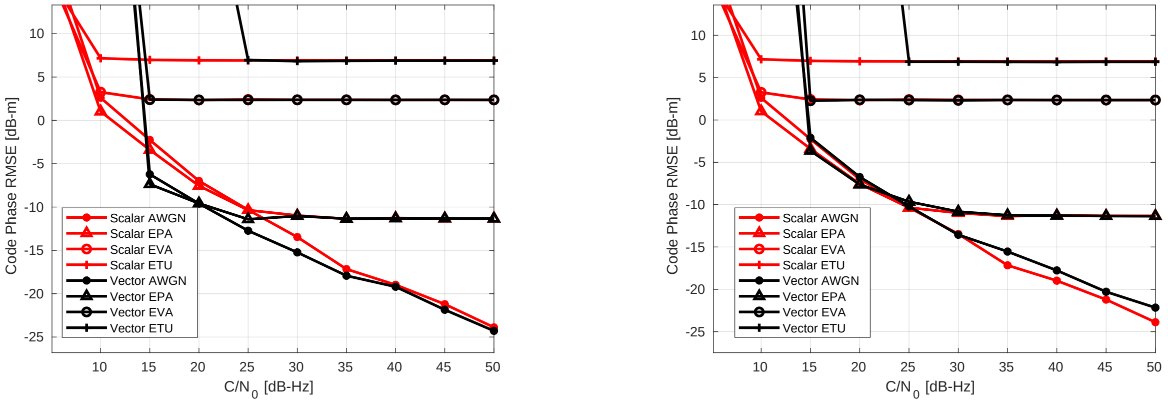

Figure 5.22: The code phase tracking performance of the 4 GPS and 1 eNodeB static simulations (left) and 5 GPS and 1 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS

lock in numerous GPS channels as well. The same problem is not present in scalar tracking.

GPS and 4 eNodeB Simulations: Figure 5.24 shows the code phase tracking RMSE of the 4 eNodeB simulations, where the LTE signal is tracked using the SSS. It is shown that, during the static simulations, the tracking RMSE is still improved over scalar tracking. However, in the dynamic simulations, the tracking begins to fail in the EVA and ETU channels. This can further be seen in Figure 5.25, where the probability of tracking the LTE signals throughout the simulations is shown. The probability of tracking during the static simulations is somewhat better than the 1 eNodeB case shown in Figure 5.21. However, in the dynamic case, the signals are tracked better in low C/N_0, but the tracking tends to break down at higher C/N_0 for the EVA and ETU channels.

Figure 5.26 shows the results of the simulations where the LTE signal is tracked using the CRS. The results indicate that the CRS is tracked at least as well as scalar tracking in the static and dynamic simulations. There is still a constant error incurred, but it is much less than the error caused by tracking the SSS even in the EPA channel. Figure 5.27 shows the probability of tracking the LTE signal during the simulations. It is shown that the tracking is

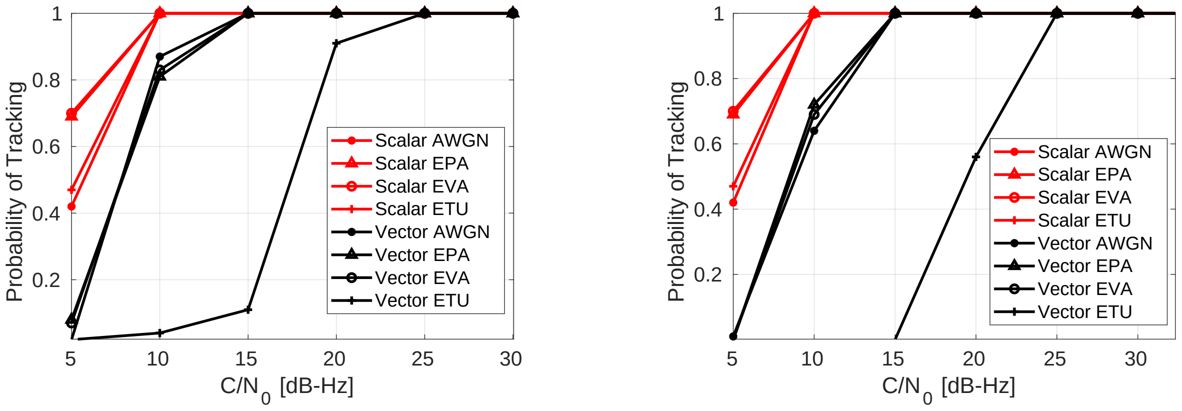

Figure 5.23: The probability of tracking the LTE signal during the 4 GPS and 1 eNodeB static simulations (left) and 5 GPS and 1 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS.

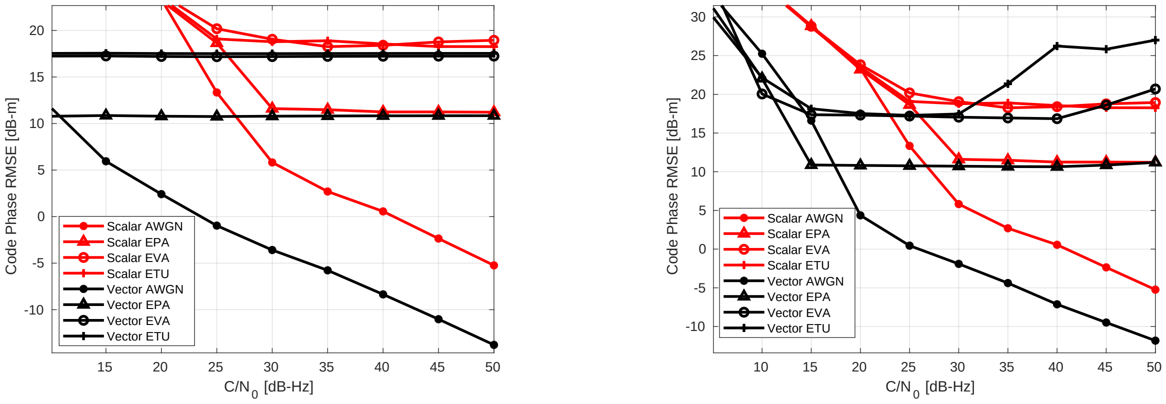

Figure 5.24: The code phase tracking performance of the 4 GPS and 4 eNodeB static simulations (left) and 5 GPS and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS

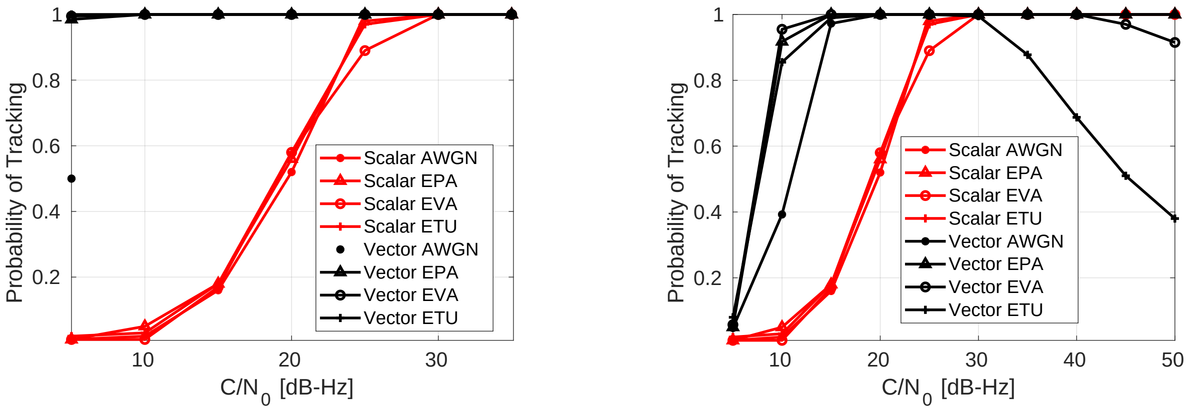

Figure 5.25: The probability of tracking the LTE signal during the 4 GPS and 4 eNodeB static simulations (left) and 5 GPS and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS.

comparable to the 1 eNodeB case presented in Figure 5.23. The LTE signal is still lost early in the ETU channel. Finally, the 4 eNodeB simulations are performed for both scenarios. The results of these simulations are shown in Appendix B. The largest observation is that the GPS signals aid in tracking the LTE signals in the multipath channels. The SSS signals in the EVA and ETU channels are lost earlier. The CRS signals in the EVA channel begin to waiver as well when the C/N_0 becomes higher.

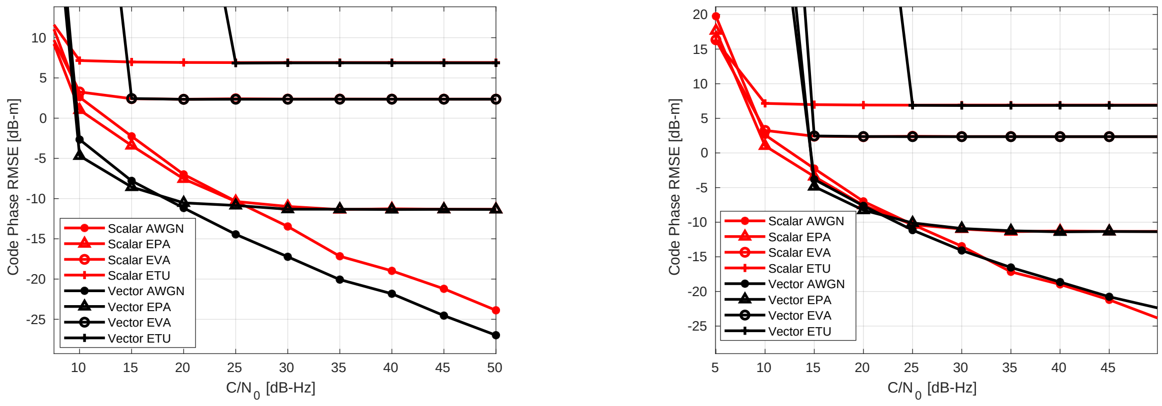

Figure 5.26: The code phase tracking performance of the 4 GPS and 4 eNodeB static simulations (left) and 5 GPS and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS

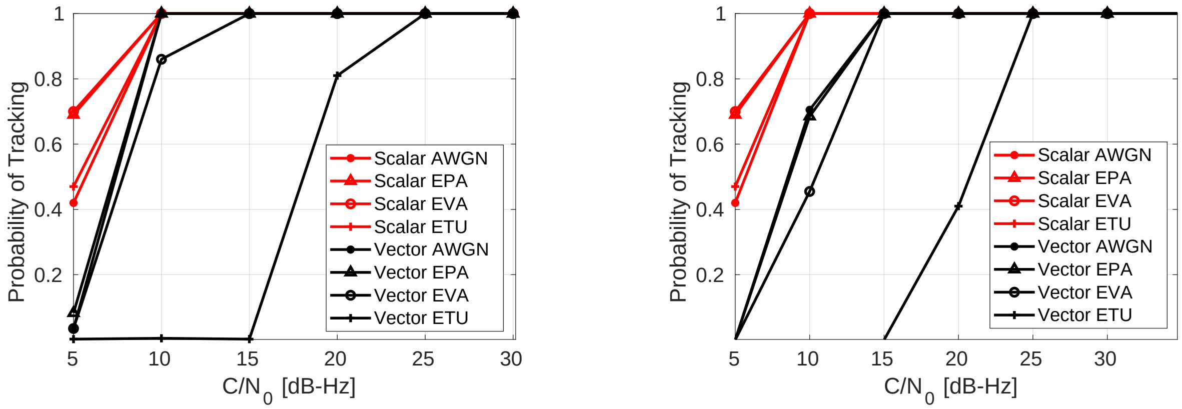

Figure 5.27: The probability of tracking the LTE signal during the 4 GPS and 4 eNodeB static simulations (left) and 5 GPS and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS.

Outage Simulations

Finally, the results of the outage simulations are shown in Figure 5.28. Each simulation scenario is subjected to both multipath and eNodeB localization error. When at least 3 LTE signals are tracked using the SSS through the EPA multipath channel or better the probability of tracking the GPS signals is nearly 100%. With 1 eNodeB present, the probability of tracking is higher than the GPS-only VDFLL and scalar tracking for all scenarios. Once 2 eNodeBs are present, the probability of tracking is better for all scenarios except when the LTE signals are subjected to the ETU channel. Once there are three or more present, the ETU channel actually reduces the probability of tracking of the GPS signals. However, regardless of channel and eNodeB localization error, the GPS signal is tracked with higher probability than the GPS-only VDFLL and scalar tracking. When tracking the CRS, the probability of tracking the GPS signals is always higher than the GPS-only VDFLL and scalar tracking. Once 3 eNodeBs are present, the probability of tracking the GPS signals is nearly 100% for all scenarios. With less than 3 eNodeBs the probability of tracking the GPS signals never drops below 50%.

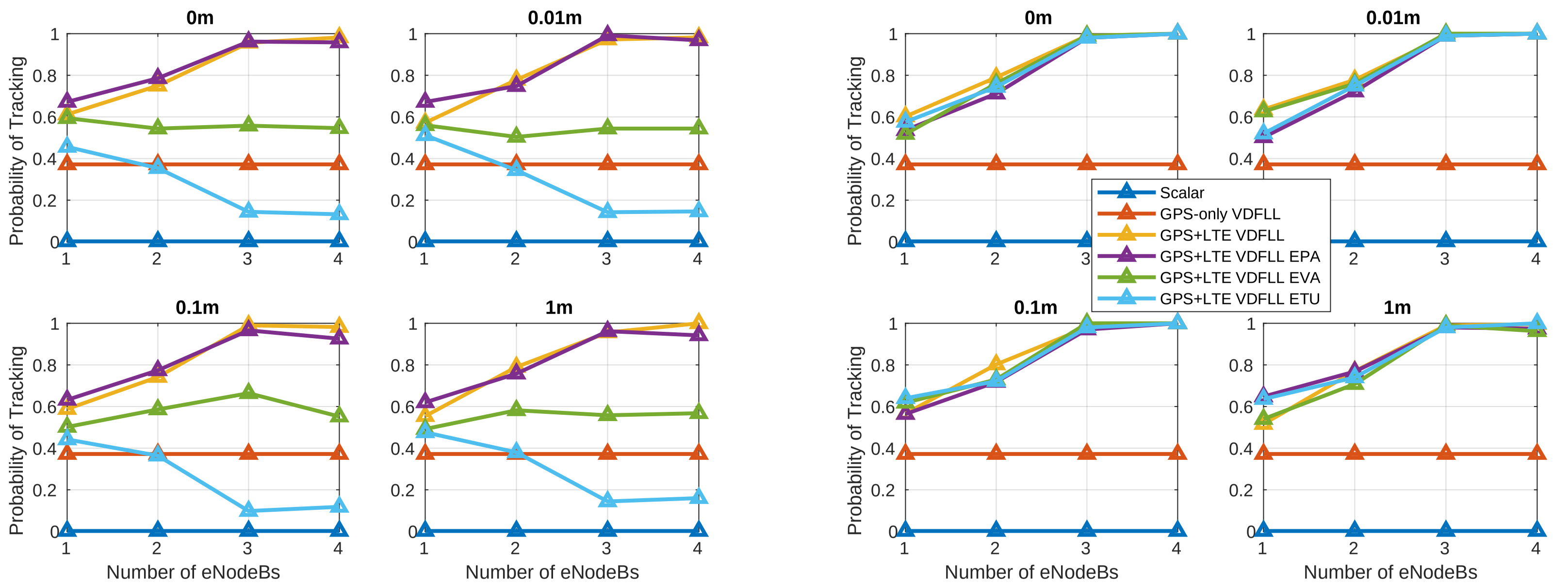

Figure 5.28: The probability of tracking the GPS signals throughout the outage simulations. On the left, the SSS is tracked. On the right, the CRS is tracked.

5.2 Experiments

5.2.1 Experimental Setup

To validate the feasibility of the combined VDFLL, real signal data are collected. To do so, a data collection setup is contrived. Figure 5.29 shows the experimental setup. The data are collected in a Lincoln MKZ. The MKZ is outfitted with a NovAtel Pinwheel antenna and a Molex multi-hub 5-in-1 GPS, LTE, and Wi-Fi antenna. The Pinwheel antenna is fed to a Honeywell eTalin. The eTalin provides a military-grade GPS/INS solution. The solution is used as ground truth for the experiments. The Molex antenna is fed to an Ettus Research USRP X310. The GPS feed is split. The GPS data are sampled at 25 MSPS and fed to an onboard GPSDO to discipline the USRP clock. One or both of the LTE feeds are used, depending on the experiment. The LTE data are sampled at 25 MSPS as well. The raw signal data are recorded by a host computer to an external solid-state drive. The raw signal data are then processed using the combined VDFLL described in Chapter 4.

There are 3 experiments performed in this thesis. There are 2 data collections performed near the Thomas Walter MRI building in Auburn, Al; one is static and the other is dynamic. Figure 5.30 shows the static and dynamic ground truths for these experiments. GPS and LTE data are collected in both experiments. The LTE data comes from eNodeB 1 in Figure 5.30 at a center frequency of 885 MHz. The GPS satellites are the same as in Figure 5.2 for the static and dynamic data collections, respectively. Both experiments last approximately 120 seconds.

The final experiment is performed in Atlanta, GA. Figure 5.31 shows the ground truth trajectory and the eNodeBs present during the experiment. During this experiment, signal data from 4 GPS satellites are recorded. Data from 2 eNodeBs (eNodeB 1 and eNodeB 2 in Figure 5.31) at center frequencies of 751 and 1980 MHz, respectively are also recorded. Figure 5.32 shows the skyplot of the present GPS satellites. Notably, the GPS geometry is fairly poor. Table 5.5 shows the PDOPs of the GPS-only and GPS and LTE recordings. It

Figure 5.29: The equipment setup in the data collection vehicle. The GPS/LTE antenna is connected to two channels on the USRP X310. The GPS output is split and is fed to an onboard GPSDO in the USRP X310. The Pinwheel antenna is connected to the eTALIN for truth.

is shown that by adding the 2 eNodeBs the geometry is improved significantly. It should be noted that this experiment is much shorter than the previous one. This is because of limitations imposed by the recording setup. By recording three complex, 8-bit, channels at sampling rates of 15 to 25 MSPS at once, recordings were limited to less than 1 minute. Some methods to resolve this are considered, later.

Table 5.5: The PDOPs of the scenarios recorded in Atlanta, GA.

GPS	**GPS + LTE**
97.66	2.59

GPS/LTE VDFLL Modifications for Experimental Data

As noted previously, the VDFLL is essentially an EKF at its core. The EKF makes some key assumptions about the measurements it uses. Primarily, that the measurement

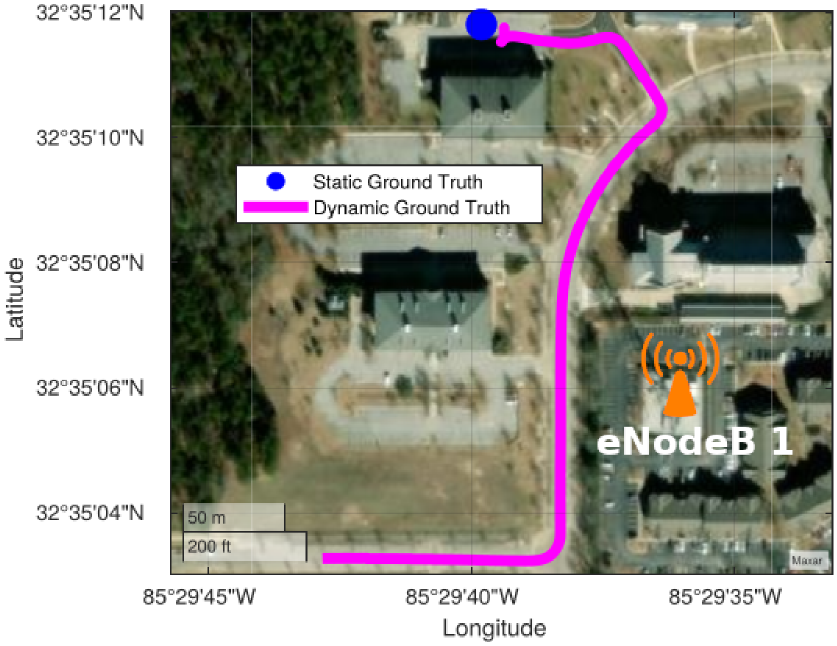

Figure 5.30: The ground truth static and dynamic trajectories for the experimental data recorded in Auburn, AL.

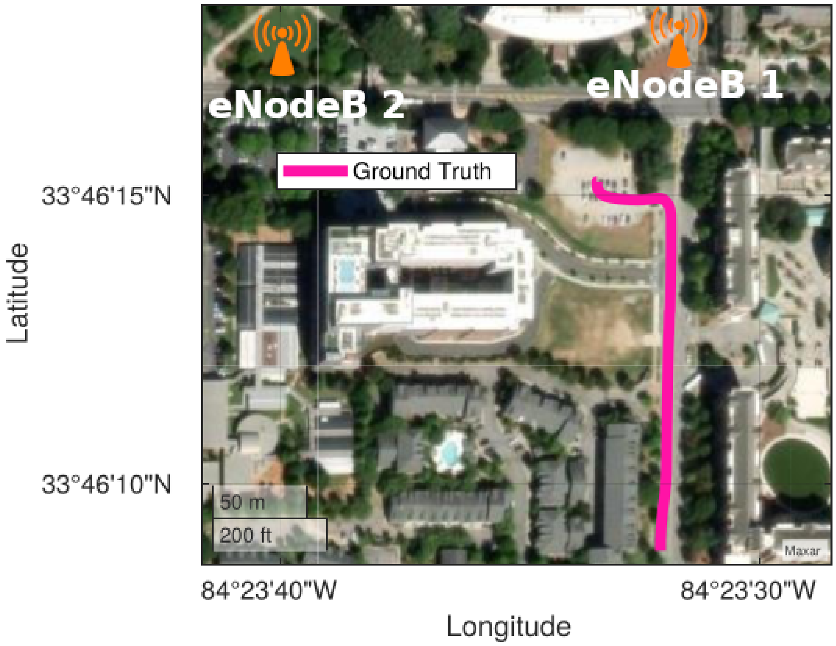

Figure 5.31: The ground truth trajectory for the experimental data recorded in Atlanta, GA.

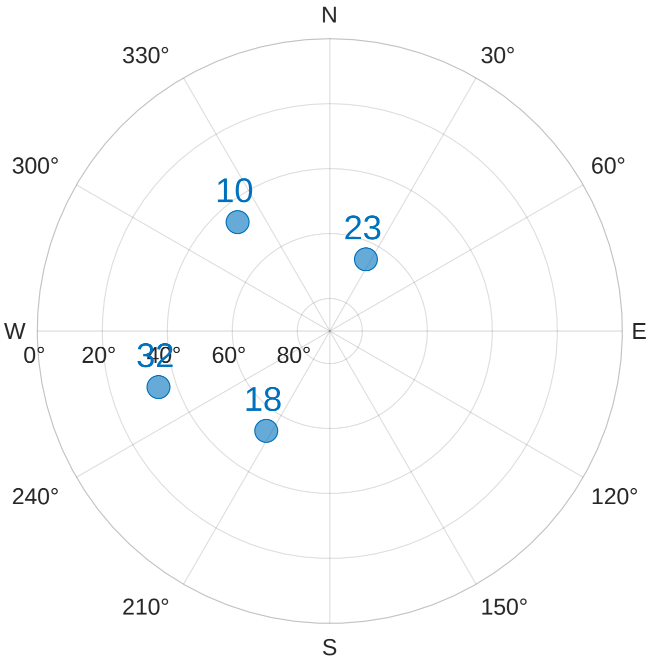

Figure 5.32: The skyplot of the GPS satellites that are tracked in the experimental data recorded in Atlanta, GA.

residuals are drawn from a zero mean multivariate normal distribution with covariance $\mathbf{R}$. Since this covariance is provided by the receiver as a function of C/N_0, the performance of the EKF depends on the accuracy of this provision. If the provided $\mathbf{R}$ is too small the filter will have too much confidence in inaccurate measurements, causing a degraded solution. With $\mathbf{R}$ too large, the filter will trust the dynamic model too much. This will cause the solution to wander aimlessly, depending on the quality of the dynamic model. Indeed, $\mathbf{R}$ is an important parameter and it should be selected with care.

In the simulation results presented before, the factors of eNodeB localization error and multipath are explored. It is found that these parameters caused biases in the measurements that violate the zero mean assumption. As a result, the solutions degraded accordingly. If a constant bias were present, an additional term could be added to the measurement model to account for it. Unfortunately, this would likely cause the measurement to add little to no information to the system because it would be used primarily to estimate its own bias. If all the measurements shared the same bias, this method could work. Unfortunately, unlike the simulations, it is unlikely that all measurements would be subjected to the same level of eNodeB localization error and/or multipath. To resolve this issue, much more sophisticated methods must be used. For example, a simultaneous localization and mapping method, such as what is used in [72] could be used. To mitigate the effects of multipath, a multipath estimating delay lock loop, such as the one developed in [73] could be fed forward to the VDFLL. These methods are outside the scope of this thesis. Instead, the VDFLL EKF will simply trust the measurements less. That is, the measurement covariance of the LTE measurements will be heuristically scaled until the best performance is achieved. For the data collected in this thesis, the covariances shown in equations (4.14) and (4.15) are scaled by 50 and 100 for the SSS and CRS, respectively. This way, some performance gain can be shown without compromising the solution.

5.2.2 Experimental Results

Auburn Experiments

In the first experiment, the data collection vehicle remains static while the GPS and LTE signal data are recorded. First, the GPS data are processed using a scalar receiver. The GPS ephemeris data is decoded and recorded for later use. It is also the same ephemeris data used in the static simulations earlier. The LTE data are recorded from eNodeB 1 in Figure 5.30. Some system information must first be decoded from the LTE signal before it is used for navigation. From inspection, it is determined to have a bandwidth of 10 MHz. Therefore, the LTE signal data are decimated from 25 MSPS to 15.36 MSPS, according to Table 4.1. Next, the LTE signal data are processed in the scalar receiver described in Chapter 3. The LTE signal from eNodeB 1 is determined to have a cell ID of 330. With this information, it is possible to produce a valid replica of the SSS and CRS. First, the issue of the eNodeB clock states must be considered. The eNodeB clocks are assumed to be relatively stable over the short term, so an estimate of the eNodeB's clock drift is determined using scalar tracking. This clock drift is determined to be 0.0126 m/s and assumed to be constant for the duration of the experiment. This clock drift and its integral are used within the VDFLL, later.

The GPS/LTE VDFLL begins processing at the first decoded GPS subframe. The LTE signals are reacquired at the latest decoded GPS subframe. Then, the GPS/LTE VDFLL begins to operate as normal. The GPS signal data are processed using 20 ms predetection integration periods and the LTE data are processed in 20 ms intervals as well. However, the SSS only appears twice within a 20 ms period, so the actual integration period is much smaller. Over 20 ms the CRS occurs with $N = 80$ times. The symbol timing and frequency error is assumed constant over 20 ms. Hence, 80 separate discriminators are calculated and averaged. The determined clock drift and its integral are used in the pseudorange rate and pseudorange estimation, respectively within the GPS/LTE VDFLL.

Figure 5.33 shows the positioning performance of the combined GPS/LTE VDFLLs with the SSS and CRS compared with the GPS-only VDFLL. It is shown from inspection that the GPS + CRS VDFLL has the best performance and the GPS + SSS VDFLL has the second best. The GPS-only VDFLL exhibits the worst performance, which is unsurprising because of the GPS-only PDOP shown in Table 5.2. The 3D statistics for the static experiment are shown in Table 5.6. It is shown that the addition of LTE improves the 3D RMSE by 28% to 34% with the SSS and CRS, respectively. The standard deviation is improved by 30% to 35% with the SSS and CRS, respectively. The max 3D error is decreased by 30% to 35% with the SSS and CRS, respectively. Figure 5.34 shows the errors in the east, north, and up axes. It is shown that the GPS-only error primarily diverges from the others on the north axis.

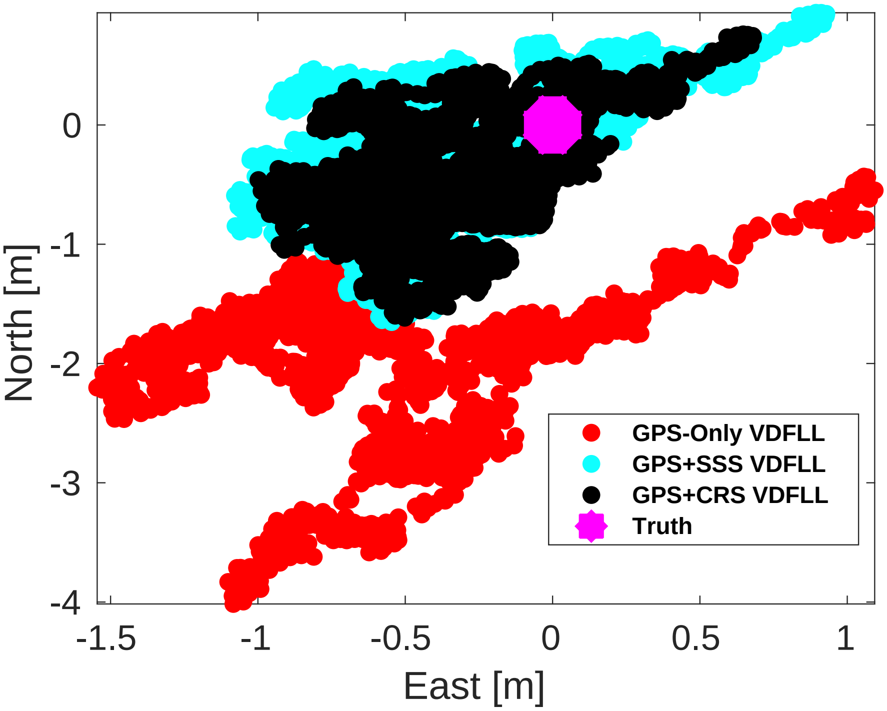

Figure 5.33: Positioning performance of the GPS-only VDFLL and combined VDFLLs using the static data recorded in Auburn, AL.

Next, the data from the dynamic route in Figure 5.30 are processed. The data are processed similarly to the static run, but this time, the eNodeB clock drift is determined to

Table 5.6: Static Experiment Statistics

Metric	**GPS**	**GPS + SSS**	**GPS + CRS**
RMSE (m)	3.00	2.17	1.99
Standard Deviation (m)	1.36	0.95	0.88
Maximum Error (m)	5.71	4.01	3.73

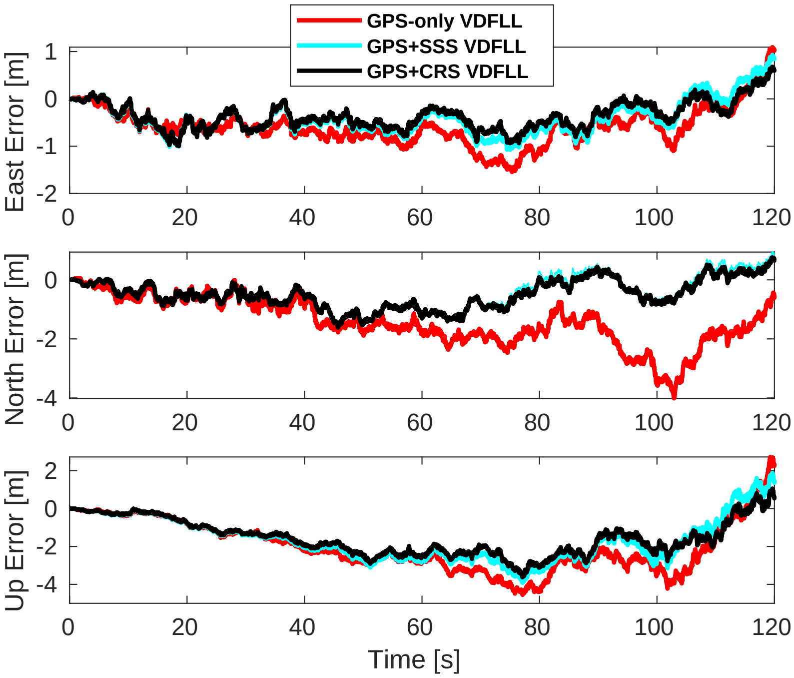

Figure 5.34: The position errors in the east, north, and up axes for the static data collected in Auburn, AL.

be 0.06 m/s. Figure 5.35 shows the results of this run. It is shown that all of the VDFLLs perform similarly. However, the VDFLLs that include the LTE signals begin to outclass the GPS-only VDFLL. Figure 5.36 shows the east, north, and up errors of the run. In this particular run, it is clear that the addition of the LTE signals provides a benefit, particularly in the east axis. The GPS + SSS and GPS + CRS VDFLL perform similarly, but the GPS + SSS VDFLL performs the best overall. Table 5.7 shows the statistics of the dynamic run. The 3D RMSE improves by 21% and 15% with the GPS + SSS and GPS + CRS VDFLLs, respectively. The 3D standard deviation improves by 21% and 17%. The maximum error improves by 9% with the GPS + SSS VDFLL. However, the maximum error of the GPS + CRS VDFLL degrades by around 2%. The GPS + SSS VDFLL outperforming the GPS +

CRS VDFLL is surprising but not impossible. Given the simulation results, it is possible that, given poor eNodeB localization and low multipath, tracking the LTE signal with the SSS may outperform the CRS. Not to mention, the effect would be more pronounced during the dynamic run than the static run. Additionally, the environment in Auburn, AL is not particularly urban and would not provide a very harsh multipath channel.

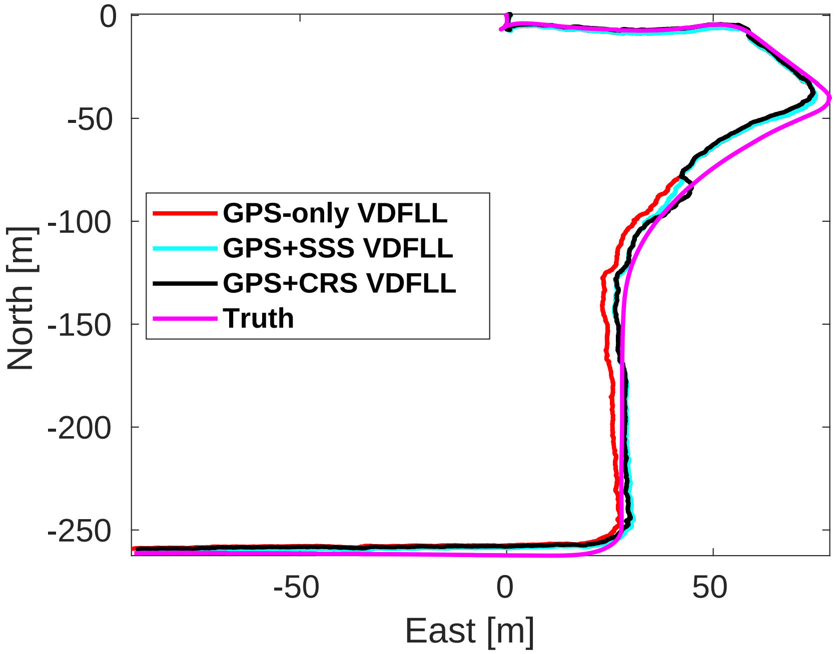

Figure 5.35: Positioning performance of the GPS-only VDFLL and the combined VDFLLs using the dynamic data recorded in Auburn, AL.

Table 5.7: Dynamic Experiment Statistics

Metric	**GPS**	**GPS + SSS**	**GPS + CRS**
RMSE (m)	4.06	3.20	3.44
Standard Deviation (m)	1.76	1.39	1.59
Maximum Error (m)	7.32	6.69	7.46

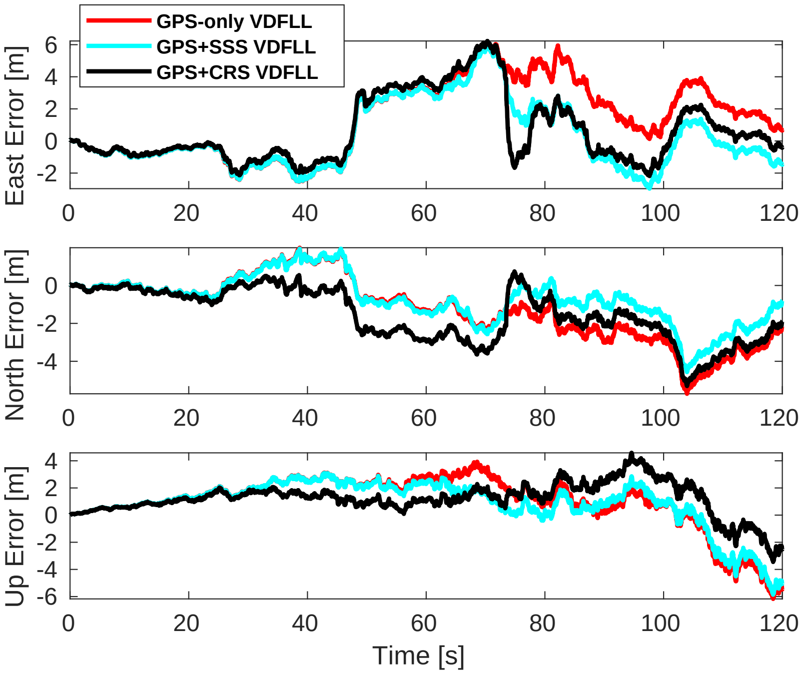

Figure 5.36: The position errors in the east, north, and up axes for the dynamic data collected in Auburn, AL.

Atlanta Experiment

Finally, the data recorded in Atlanta, GA are processed. This data features 2 eNodeBs. The data are processed in the same way as above. Both eNodeBs are determined by inspection to have a bandwidth of 10 MHz. The LTE signals are acquired and are noted to have cell IDs of 411 and 217 for eNodeB 1 and eNodeB 2, respectively. The clock drifts are determined to be 0.2 m/s and 0.22 m/s for eNodeB 1 and eNodeB 2, respectively. Finally, the data are processed in the same way. Once with just the GPS signal data and twice with the GPS and LTE signal data. The LTE signal data are tracked using the SSS and the CRS in two separate runs. Figure 5.37 shows the results of processing the dynamic data recorded in Atlanta, GA. It is shown that the GPS-only and GPS + SSS VDFLLs perform nearly equivalently. The GPS + CRS VDFLL initially performs the worst but eventually overtakes the other VDFLLs. Figure 5.38 shows the errors in the east, north, and up axes. It is clear

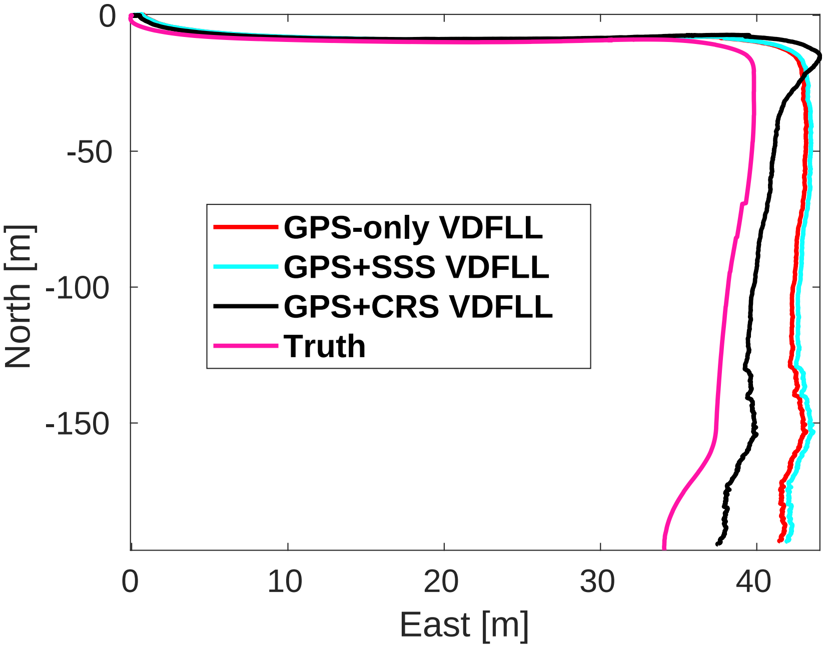

Figure 5.37: Positioning performance of the GPS-only VDFLL and the combined VDFLLs using the dynamic data recorded in Atlanta, GA.

that the GPS + CRS VDFLL easily overtakes the other VDFLL implementations. In particular, the error in the up-axis is improved considerably. Table 5.8 summarizes the statistics of the run. In this run, tracking the LTE signals with the SSS degraded the performance of the VDFLL. The 3D RMSE, standard deviation, and maximum error are all degraded by 45%, 49%, and 46%, respectively. In contrast, tracking the LTE signals with the CRS provided a large performance increase. The 3D RMSE, standard deviation, and maximum error are all improved by 29%, 8%, and 13%, respectively. Given that the data recorded in Atlanta is more likely to be severely affected by multipath, this result is not surprising. While the eNodeB localization is also not exact, the combination causes the SSS to only be destructive to the solution.

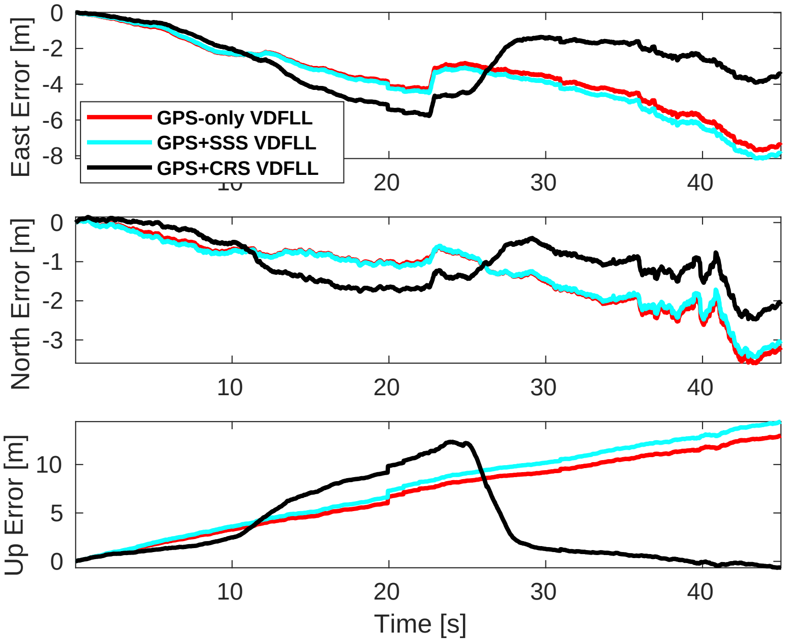

Figure 5.38: The position errors in the east, north, and up axes for the dynamic data collected in Atlanta, AL.

Table 5.8: Atlanta Statistics

Metric	**GPS**	**GPS + SSS**	**GPS + CRS**
RMSE (m)	8.97	12.97	6.36
Standard Deviation (m)	4.24	6.34	3.92
Maximum Error (m)	15.30	22.33	13.26

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, a GPS L1 C/A and Cellular 4G LTE vector tracking receiver is developed. The signal models of the GPS L1 C/A and 4G LTE signals are described. Then, the traditional, scalar tracking loops are discussed, outlining the common discriminators and loop filters used to track the GPS and LTE signals. Positioning with measurements derived from RF signal tracking is briefly treated as well. Next, these concepts are extended and combined in a discussion about vector tracking loops and the VDFLL. The combined VDFLL is then developed and the pseudorange and pseudorange rate residual covariances are given in terms of C/N_0. To use these effectively, a method of calculating the C/N_0 using the LTE tracking correlator outputs is developed as well.

The performance of the combined VDFLL is evaluated using DARE and Monte Carlo simulation. The Monte Carlo simulations match reasonably well with the DARE results, proving that the covariance calculations and C/N_0 calculations developed in this thesis are accurate. The DARE and Monte Carlo results also show that the code and carrier tracking performance can be improved significantly by tracking LTE signals in a combined VDFLL. The more LTE signals tracked, the better the solution would become. In addition, it is found that tracking the LTE signals with the CRS provides better performance than using the SSS. It is also shown that the CRS is particularly vulnerable to disturbances such as eNodeB localization error and multipath due to its sample-level tracking requirements and low covariance. It is found that the SSS is relatively robust against eNodeB localization error, but it is very vulnerable to multipath. If either the SSS or CRS were subjected to these errors and no steps were taken to mitigate their errors, the overall position solution would be degraded. In addition, it is shown that including LTE signals aided in tracking the GPS signals through outage periods in a dynamic simulation regardless of additional error when tracking the CRS. If the multipath was not severe, the SSS also provided some benefit over the GPS-only VDFLL.

Finally, real signal data were collected and processed using the combined VDFLL. Data were collected in Auburn, AL, and Atlanta GA. Static and dynamic data were collected. To mitigate the possible degradation caused by multipath and eNodeB localization error, the covariances of the LTE measurements are inflated. Heuristically, it is found that the SSS covariances should be scaled by 50 and the CRS covariances should be scaled by 100. In all experiments, the combined VDFLL performs better than the traditional GPS-only VDFLL. In Auburn, the 3D positioning RMSE was improved by 28% and 34% with the SSS and CRS, respectively for the static experiment. In the dynamic experiment, the 3D positioning RMSE was improved by 21% and 15% with the SSS and CRS, respectively. In Atlanta, the 3D positioning RMSE was improved by 29% when tracking the LTE signals with the CRS. However, by tracking the SSS the position solution was degraded.

Overall, the combined VDFLL shows promise in both simulation and experimentation. The CRS shows the most promise in terms of positioning accuracy, but it is the most vulnerable to degradation. The SSS is more robust, but it is more susceptible to multipath. If other disturbances are large, both the SSS and CRS will degrade the solution. To resolve this, some steps should be taken. Some possibilities will be discussed next.

6.2 Future Work

As discovered in simulation, eNodeB localization and multipath are large sources of error. They can cause significant positioning errors and, if they are severe enough, loss of lock. The first steps taken should be to mitigate these effects. One such method could be to precisely localize the broadcasting eNodeBs. To do so, one may place multiple receivers with known positions around the eNodeB of interest. A similar method of extracting the eNodeB clock states as used in this thesis could be employed here as well. Then, the inverse problem could be solved for the eNodeB position. Unfortunately, this method would also be meter level at best. To resolve this, a long period of data should be collected. The resultant solutions could be averaged until a desired level of precision is determined. Yet, this is just one way to solve this problem. Other approaches using simultaneous localization and mapping could be used instead to varying degrees of accuracy.

Resolving the issue of multipath is a more difficult problem. Multipath still remains a troublesome issue even in GNSS signal tracking, where it is less predominant than in LTE signal tracking. Numerous methods of mitigating multipath exist in the literature. For example, one may add the multipath bias into the pseudorange measurement model. This is only effective if each signal is subjected to the same multipath channel. This is unlikely. Other methods use additional correlators within a traditional DLL to mitigate this error. It is possible that this could be fed forward to the VDFLL code frequency estimate.

Another possible avenue of research could be to couple the combined VDFLL with another sensor. The most obvious of which being an INS. Yet, a heading reference alone would probably provide some performance gain. Other sensors that could provide redundant information such as a camera, lidar, or radar would also improve the solution. Moreover, combining the receiver with additional GNSS or other SOPs would be worthwhile as well.

Finally, it is worth mentioning the recording setup. It was noted that when recording more than two channels, only around 40 seconds of continuous data could be recorded. A method of improving this recording setup could be investigated. This would allow for more LTE signals to be used in the VDFLL, potentially improving the solution. To do so, the most obvious solution would be to write to faster memory. While the solid state drive used in this work purports fast write speeds, the overhead of writing multiple complex channels at rates of at least 15 MSPS each is too much. A possible solution, given the host computer has enough Random Access Memory (RAM), would be to write to a so-called 'RAM disk'. This method would write directly to the much faster, but volatile system memory. However, this method has its pitfalls as well. For one, the host PC would need to have 10s of gigabytes of *unused* RAM, which is uncommon, especially in a portable PC.

Ultimately, given that the combined VDFLL shows promise using the basic methods provided in this thesis, large performance gains are likely possible. The simulations performed in this work show that these gains are possible in AWGN conditions. By implementing any one or all of these methods the performance gains shown by the simulations performed in this work could possibly be achieved or, at least, approached.

Bibliography

[1] P. Xie and M. G. Petovello, "Measuring GNSS Multipath Distributions in Urban Canyon Environments," in *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 2, pp. 366-377, Feb. 2015, doi: 10.1109/TIM.2014.2342452.

[2] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects," in *Information Fusion*, vol. 7, no. 2, pp. 221–230, Jun. 2006, doi: 10.1016/j.inffus.2004.07.002.

[3] X. Meng, H. Wang, and B. Liu, "A Robust Vehicle Localization Approach Based on GNSS/IMU/DMI/LiDAR Sensor Fusion for Autonomous Vehicles," in *Sensors*, vol. 17, no. 9, pp. 2140-2159, Sep. 2017, doi: 10.3390/s17092140.

[4] L. Wei, C. Cappelle and Y. Ruichek, "Camera/Laser/GPS Fusion Method for Vehicle Positioning Under Extended NIS-Based Sensor Validation," in *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 11, pp. 3110-3122, Nov. 2013, doi: 10.1109/TIM.2013.2265476.

[5] M. Lashley, D. M. Bevly and J. Y. Hung, "Performance Analysis of Vector Tracking Algorithms for Weak GPS Signals in High Dynamics," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 661-673, Aug. 2009, doi: 10.1109/JSTSP.2009.2023341.

[6] T. Watts, *A GPS and GLONASS L1 Vector Tracking Software-Defined Receiver*. M.S. thesis, Samuel Ginn College of Eng., Auburn Univ., Auburn, AL, 2019. [Online]. Available: https://etd.auburn.edu/handle/10415/6899

[7] C. A. Givhan, *A GPS L5 Software Defined Vector Tracking Receiver*. M.S. thesis, Samuel Ginn College of Eng., Auburn Univ., Auburn, AL, 2021. [Online]. Available: https://etd.auburn.edu/handle/10415/7943

[8] J. J. Spilker, "Vector Delay Lock Loop Processing of Radiolocation Transmitter Signals," Stanford, CA, March 1995. US Patent 5398034.

[9] D. Benson "Interference Benefits of a Vector Delay Lock Loop (VDLL) GPS Receiver," in *Proceedings of the 63rd Annual Meeting of the Institute of Navigation*, Cambridge, MA, USA 2007, pp. 749-756.

[10] M. Lashley and D. M. Bevly. "Analysis of Discriminator Based Vector Tracking Algorithms," in *Proceedings of the 2007 National Technical Meeting of the Institute of Navigation*, San Diego, CA, 2007, pp. 570-576.

[11] M. Lashley, D. M. Bevly and J. Y. Hung, ”Impact of carrier to noise power density, platform dynamics, and IMU quality on deeply integrated navigation,” in *2008 IEEE/ION Position, Location and Navigation Symposium*, Monterey, CA, USA, 2008, pp. 9-16, doi: 10.1109/PLANS.2008.4569964.

[12] M. Lashley and D. M. Bevly. ”Vector Delay/Frequency Lock Loop Implementation and Analysis,” in *Proceedings of the 2009 International Technical Meeting of the Institute of Navigation*, Anaheim, CA, USA, 2009, pp. 1073-1086.

[13] J. McEllroy, *Navigation Using Signals of Opportunity in the AM Transmission Band*. M.S. thesis, School of Eng., Air Force Inst. of Tech., Wright-Patterson Air Force Base, OH, 2006. [Online]. Available: https://scholar.afit.edu/cgi/viewcontent.cgi?article=4453&context=etd

[14] V. Moghtadaiee, A. G. Dempster and S. Lim, ”Indoor localization using FM radio signals: A fingerprinting approach,” in *2011 International Conference on Indoor Positioning and Indoor Navigation*, Guimaraes, Portugal, 2011, pp. 1-7, doi: 10.1109/IPIN.2011.6071932.

[15] P. Thevenon, S. Damien, O. Julien, C. Macabiau, M. Bousquet, L. Ries, and S. Corazza, “Positioning using mobile TV based on the DVB-SH standard,” in *NAVIGATION*, vol. 58, no. 2, pp. 71–90, Jun. 2011, doi: 10.1002/j.2161-4296.2011.tb01793.x.

[16] J. Khalife and Z. M. Kassas, ”Precise UAV navigation with cellular carrier phase measurements,” in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Monterey, CA, USA, 2018, pp. 978-989, doi: 10.1109/PLANS.2018.8373476.

[17] Z. Kassas, J. Morales, and J. Khalife, ”New-Age Satellite-Based Navigation–STAN: Simultaneous Tracking and Navigation with LEO Satellite Signals,” in *Inside GNSS Magazine*, vol. 14, no. 4, pp. 56-65, Aug. 2019.

[18] S. Mcdougal, S. Morgan, and S. Martin, ”Single-Antenna Low Earth Orbit Signal Simulator for Hardware in the Loop Testing,” in *Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022)*, Denver, CO, USA, 2022, pp. 2349-2361, doi: 10.33012/2022.18383.

[19] Z. M. Kassas and T. E. Humphreys, ”Observability Analysis of Collaborative Opportunistic Navigation With Pseudorange Measurements,” in *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 260-273, Feb. 2014, doi: 10.1109/TITS.2013.2278293.

[20] C. Yang and T. Nguyen, “Tracking and Relative Positioning with Mixed Signals of Opportunity: Radio DR with Mixed SOOP,” in *NAVIGATION*, vol. 62, no. 4, pp. 291–311, Dec. 2015, doi: 10.1002/navi.122.

[21] J. Khalife, K. Shamaei and Z. M. Kassas, ”Navigation With Cellular CDMA Signals—Part I: Signal Modeling and Software-Defined Receiver Design,” in *IEEE Transactions on Signal Processing*, vol. 66, no. 8, pp. 2191-2203, Apr. 2018, doi: 10.1109/TSP.2018.2799167.

[22] K. Shamaei, *Exploiting Cellular Signals for Navigation: 4G to 5G*, Ph.D. dissertation, Samueli Sch. of Eng., Univ. of California, Irvine, Irvine, CA, 2020. [Online]. Available: https://escholarship.org/uc/item/7nf4c00g

[23] Z. Z. M. Kassas, J. Khalife, K. Shamaei and J. Morales, "I Hear, Therefore I Know Where I Am: Compensating for GNSS Limitations with Cellular Signals," in *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 111-124, Sep. 2017, doi: 10.1109/MSP.2017.2715363.

[24] J. Khalife and Z. M. Kassas, "Navigation With Cellular CDMA Signals—Part II: Performance Analysis and Experimental Results," in *IEEE Transactions on Signal Processing*, vol. 66, no. 8, pp. 2204-2218, Apr. 2018, doi: 10.1109/TSP.2018.2799166.

[25] C. Mensing, S. Sand and A. Dammann, "Hybrid Data Fusion and Tracking for Positioning with GNSS and 3GPP-LTE," in *Interational Journal of Navigation and Observation*, vol. 2010, pp. 1-12, Aug. 2010, doi: 10.1155/2010/812945.

[26] F. Sottile, H. Wymeersch, M. A. Caceres and M. A. Spirito, "Hybrid GNSS-Terrestrial Cooperative Positioning Based on Particle Filter," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, Houston, TX, USA, 2011, pp. 1-5, doi: 10.1109/GLOCOM.2011.6134002.

[27] G. De Angelis, G. Baruffa and S. Cacopardi, "GNSS/Cellular Hybrid Positioning System for Mobile Users in Urban Scenarios," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 313-321, Mar. 2013, doi: 10.1109/TITS.2012.2215855.

[28] J. A. del Peral-Rosado, J.A. Lopez-Salcedo, G. Seco-Granados, F. Zanier, P. Crosta, R. Ioannides, and M. Crisci, "Software-Defined Radio LTE Positioning Receiver Towards Future Hybrid Localization Systems," in *31st AIAA International Communications Satellite Systems Conference*, Florence, Italy, 2013, pp. 5610-5621, 2013, doi: 10.2514/6.2013-5610.

[29] J.-J. van de Beek, O. Edfors, M. Sandell, S. K. Wilson and P. O. Borjesson, "On channel estimation in OFDM systems," in *1995 IEEE 45th Vehicular Technology Conference. Countdown to the Wireless Twenty-First Century*, Chicago, IL, USA, 1995, pp. 815-819 vol.2, doi: 10.1109/VETEC.1995.504981.

[30] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 984-995, Jul. 1989, doi: 10.1109/29.32276.

[31] M. Driusso, C. Marshall, M. Sabathy, F. Knutti, H. Mathis and F. Babich, "Vehicular Position Tracking Using LTE Signals," in *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3376-3391, Apr. 2017, doi: 10.1109/TVT.2016.2589463.

[32] M. Driusso, F. Babich, F. Knutti, M. Sabathy and C. Marshall, "Estimation and tracking of LTE signals time of arrival in a mobile multipath environment," in *2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Zagreb, Croatia, 2015, pp. 276-281, doi: 10.1109/ISPA.2015.7306072.

[33] R. Gold, "Optimal binary sequences for spread spectrum multiplexing (Corresp.)," in *IEEE Transactions on Information Theory*, vol. 13, no. 4, pp. 619-621, Oct. 1967, doi: 10.1109/TIT.1967.1054048.

[34] C. Hamm, *Analysis of simulated performance of integrated vector tracking and navigation loops for GPS*. M.S. thesis, Samuel Ginn College of Eng., Auburn Univ., Auburn, AL, 2005. [Online]. Available: https://etd.auburn.edu/xmlui/handle/10415/378

[35] 3GPP, "Evolved universal terrestrial radio access (E-UTRA); physical channels and modulation," 3rd Generation Partnership Project (3GPP), TS 36.211, January 2011. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36211.htm

[36] K. Shamaei, J. Khalife, Z. M. Kassas, "Comparative Results for Positioning with Secondary Synchronization Signal versus Cell Specific Reference Signal in LTE Systems," in *Proceedings of the 2017 International Technical Meeting of The Institute of Navigation*, Monterey, CA, 2017, pp. 1256-1268, doi: 10.33012/2017.14885.

[37] B. Sklar, "Rayleigh fading channels in mobile digital communication systems .I. Characterization," in *IEEE Communications Magazine*, vol. 35, no. 7, pp. 90-100, Jul. 1997, doi: 10.1109/35.601747.

[38] M. Speth, S. A. Fechtel, G. Fock and H. Meyr, "Optimum receiver design for wireless broad-band systems using OFDM. I," in *IEEE Transactions on Communications*, vol. 47, no. 11, pp. 1668-1677, Nov. 1999, doi: 10.1109/26.803501.

[39] B. Yang, K. B. Letaief, R. S. Cheng and Z. Cao, "Timing recovery for OFDM transmission," in *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 11, pp. 2278-2291, Nov. 2000, doi: 10.1109/49.895033.

[40] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, S. H. Jensen, *A Software-Defined GPS and Galileo Receiver: a Single-Frequency Approach*. Springer Science & Business Media, 2007. Accessed: Feb. 21, 2023. [Online]. doi: 10.1007/978-0-8176-4540-3.

[41] D. M. Akos, *A Software Radio Approach to Global Navigation Satellite System Receiver Design*. Ph.D. dissertation, Russ College of Eng. and Tech., Ohio Univ., Athens, OH, 1997. [Online]. Available: https://web.stanford.edu/group/scpnt/gpslab/pubs/theses/DennisAkosThesis97.pdf

[42] D. M. Akos, P. L. Normark, P. Enge, A. Hansson, A. Roselind, "Real-time GPS software radio receiver," in *Proceedings of the 2001 National Technical Meeting of the Institute of Navigation*, Long Beach, CA, 2001, pp. 809-816.

[43] B. Keyser, *Design and Implementation of a SoC-Based Real-Time Vector Tracking GPS Receiver*. M.S. thesis, Samuel Ginn College of Eng., Auburn Univ., Auburn, AL, 2015. [Online]. Available: http://etd.auburn.edu/handle/10415/4580

[44] M. G. Petovello, C. O'Driscoll, G. Lachapelle, D. Borio, H. Murtaza, "Architecture and Benefits of an Advanced GNSS Software Receiver," in *Journal of Global Positioning Systems*, vol. 7, no. 2, pp. 156-168, 2008.

[45] E. Kaplan, C. Hegarty, *Understanding GPS/GNSS: Principles and Applications, Third Edition*. Artech House, 2017.

[46] M. M. Sayre, *Development of a block processing carrier to noise ratio estimator for the global positioning system*. M.S. thesis, Russ College of Eng. and Tech., Ohio Univ., Athens, OH, 2003. [Online]. Available: http://rave.ohiolink.edu/etdc/view?acc_num=ohiou1071063030

[47] R. Frank, "Polyphase codes with good nonperiodic correlation properties," in *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 43-45, January 1963, doi: 10.1109/TIT.1963.1057798.

[48] D. Chu, "Polyphase codes with good periodic correlation properties (Corresp.)," in *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 531-532, July 1972, doi: 10.1109/TIT.1972.1054840.

[49] P. Ward, "Performance Comparisons Between FLL, PLL and a Novel FLL-Assisted-PLL Carrier Tracking Loop Under RF Interference Conditions," in *Proceedings of the 11th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1998)*, Nashville, TN, Sep. 1998, pp. 783-795.

[50] 3GPP, "Evolved universal terrestrial radio access (E-UTRA); multiplexing and channel coding," 3rd Generation Partnership Project (3GPP), TS 36.212, January 2010. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36212.html.

[51] Y. . -P. E. Wang and R. Ramesh, "To bite or not to bite-a study of tail bits versus tail-biting," *Proceedings of PIMRC '96 - 7th International Symposium on Personal, Indoor, and Mobile Communications*, Taipei, Taiwan, 1996, pp. 317-321 vol.2, doi: 10.1109/PIMRC.1996.567407.

[52] F. Knutti, M. Sabathy, M. Driusso, H. Mathis, C. Marshall, "Positioning using LTE signals," *Proceedings of Navigation Conference in Europe*, Bordeaux, France, Apr. 2015, pp. 1-8.

[53] C. R. Berger, S. Zhou, J. C. Preisig and P. Willett, "Sparse Channel Estimation for Multicarrier Underwater Acoustic Communication: From Subspace Methods to Compressed Sensing," in *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1708-1721, March 2010, doi: 10.1109/TSP.2009.2038424.

[54] B. Yang, K. B. Letaief, R. S. Cheng and Z. Cao, "Channel estimation for OFDM transmission in multipath fading channels based on parametric channel modeling," in *IEEE Transactions on Communications*, vol. 49, no. 3, pp. 467-479, March 2001, doi: 10.1109/26.911454.

[55] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 33, no. 2, pp. 387-392, April 1985, doi: 10.1109/TASSP.1985.1164557.

[56] J. A. Klobuchar, "Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-23, no. 3, pp. 325-331, May 1987, doi: 10.1109/TAES.1987.310829.

[57] B. W. Parkinson, "OVERVIEW (GPS SYSTEM DESCRIPTION)", in *NAVIGATION: Journal of The Institute of Navigation*, vol. 25, no. 2, pp. 93-94, June 1978. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=100676

[58] F. Kleijer, *Troposphere modeling and filtering for precise GPS leveling.* Ph.D. dissertation, Delft Univ. of Tech., Delft, NL, 2004. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid:ea1f0cf0-4e48-421b-b7ae-4ae3e36d1880?collection=research

[59] M. Psiaki and H. Jung, "Extended Kalman filter methods for tracking weak GPS signals," in *Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002)*, Portland, OR, Sep. 2002, pp. 2539-2553. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=2278

[60] T. Pany and B. Eissfeller, "Use of a Vector Delay Lock Loop Receiver for GNSS Signal Power Analysis in Bad Signal Conditions," in *2006 IEEE/ION Position, Location, And Navigation Symposium*, Coronado, CA, USA, 2006, pp. 893-903, doi: 10.1109/PLANS.2006.1650689.

[61] J. Liu, X. Cui, M. Lu and Z. Feng, "Vector tracking loops in GNSS receivers for dynamic weak signals," in *Journal of Systems Engineering and Electronics*, vol. 24, no. 3, pp. 349-364, June 2013, doi: 10.1109/JSEE.2013.00044.

[62] Y. Luo, Y. Wang, S. Wu, P. Wang, "Multipath effects on vector tracking algorithm for GNSS signal," in *Science China Information Sciences*, vol. 57, no. 10, pp. 1–13, Sep. 2014 https://doi.org/10.1007/s11432-014-5153-1

[63] L. Hsu, S. Jan, P. Groves and N. Kubo, "Multipath mitigation and NLOS detection using vector tracking in urban environments," in *GPS Solutions*, vol. 19, no. 2, pp. 249-262, June 2014, doi: https://doi.org/10.1007/s10291-014-0384-6.

[64] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, Estimation with Applications to Tracking and Navigation. New York: John Wiley and Sons, 2001.

[65] R. G. Brown, P. Y. C. Hwang, Introduction to Random Signals and Applied Kalman Filtering With MATLAB Exercises. New York: John Wiley and Sons, 2012.

[66] R. Crane, "A simplified method for deep coupling of GPS and inertial data," in *Proceedings of the 2007 National Technical Meeting of the Institute of Navigation*, San Diego, CA, USA, 2007, pp. 311-319. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=7162

[67] M. Lashley, *Modeling and performance analysis of GPS vector tracking algorithms.* Ph.D. dissertation, Samuel Ginn Col. of Eng., Auburn Univ., Auburn, AL

[68] Xiaodong Xu, Ya Jing and Xiaohu Yu, ”Subspace-based noise variance and SNR estimation for OFDM systems [mobile radio applications],” in *IEEE Wireless Communications and Networking Conference*, 2005, New Orleans, LA, 2005, pp. 23-26 Vol. 1, doi: 10.1109/WCNC.2005.1424470.

[69] 3GPP, ”Evolved universal terrestrial radio access (E-UTRA); user equipment (UE) radio transmission and reception,” 3rd Generation Partnership Project (3GPP), TS 36.101, April 2017. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36101.html

[70] S. Morgan and S. Martin, ”Performance analysis of a cellular LTE and GPS L1 C/A vector tracking receiver, a simulation study,” in *Proceedings of the 2023 International Technical Meeting of The Institute of Navigation*, Long Beach, CA, USA, 2023, pp. 692-707, doi: 10.33012/2023.18663.

[71] S. Morgan, J. T. Koza and S. Martin, (in press) ”Experimental performance of a cellular LTE and GPS L1 C/A vector tracking receiver,” in *Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Denver, CO, USA, 2023.

[72] J. J. Morales and Z. M. Kassas, ”Optimal Collaborative Mapping of Terrestrial Transmitters: Receiver Placement and Performance Characterization,” in IEEE Transactions on Aerospace and Electronic Systems, vol. 54, no. 2, pp. 992-1007, April 2018, doi: 10.1109/TAES.2017.2773238.

[73] P. Wang and Y. J. Morton, ”Multipath Estimating Delay Lock Loop for LTE Signal TOA Estimation in Indoor and Urban Environments,” in IEEE Transactions on Wireless Communications, vol. 19, no. 8, pp. 5518-5530, Aug. 2020, doi: 10.1109/TWC.2020.2994037.

Appendices

Appendix A

4 eNodeB AWGN Simulation Results

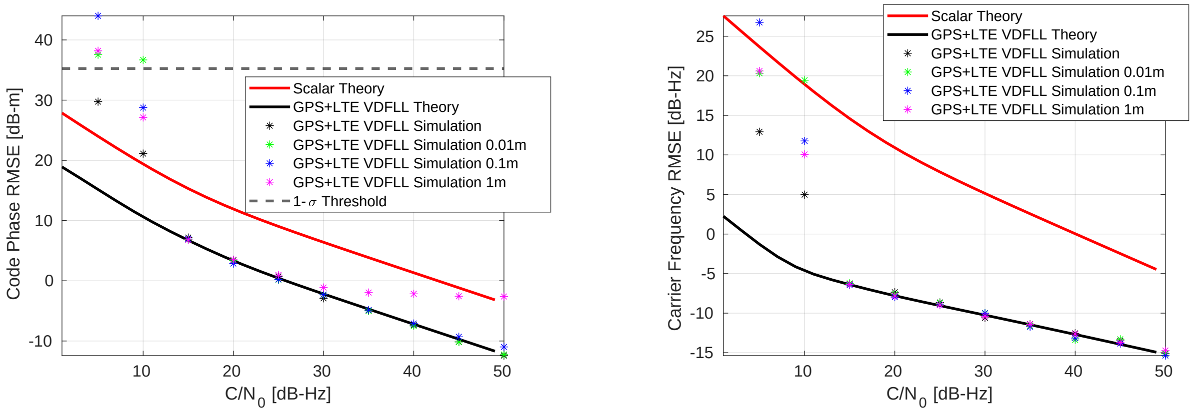

Figure A.1: 4 eNodeB static simulation results. The LTE signals are tracked using the SSS.

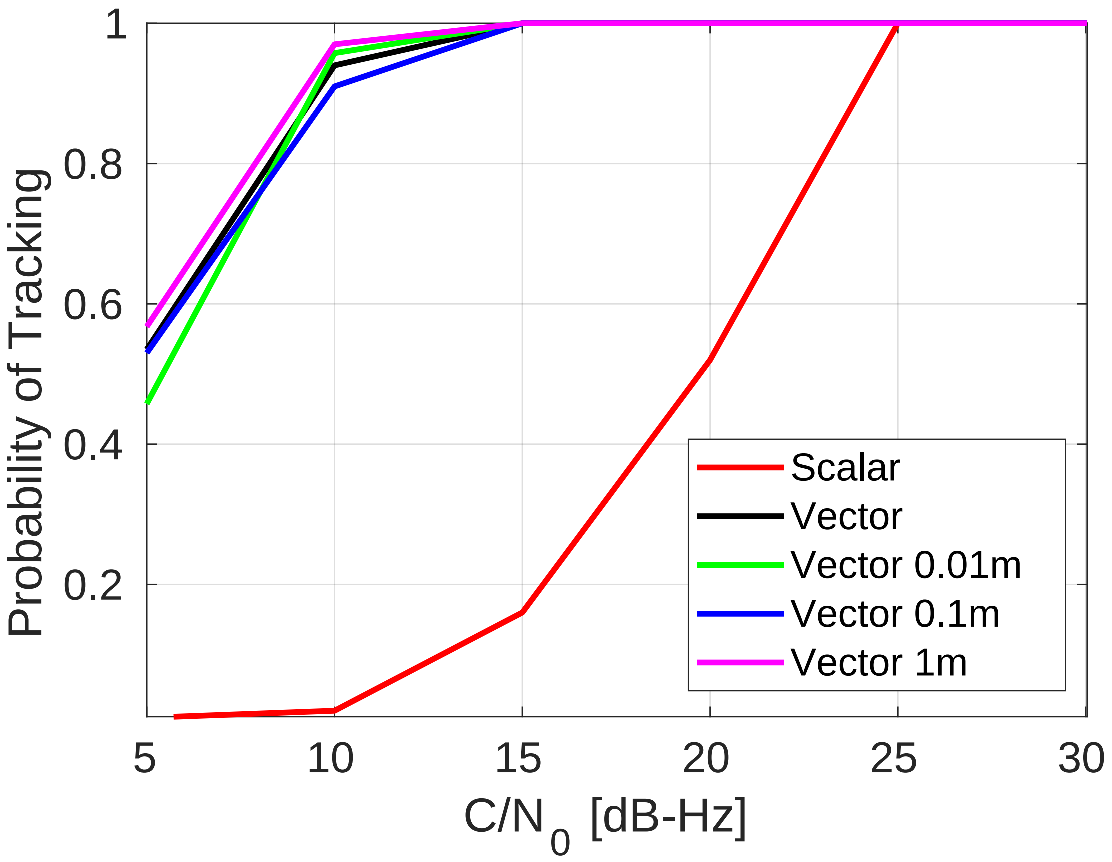

Figure A.2: Probability of tracking LTE in the 4 eNodeB static simulations. The LTE signals are tracked using the SSS.

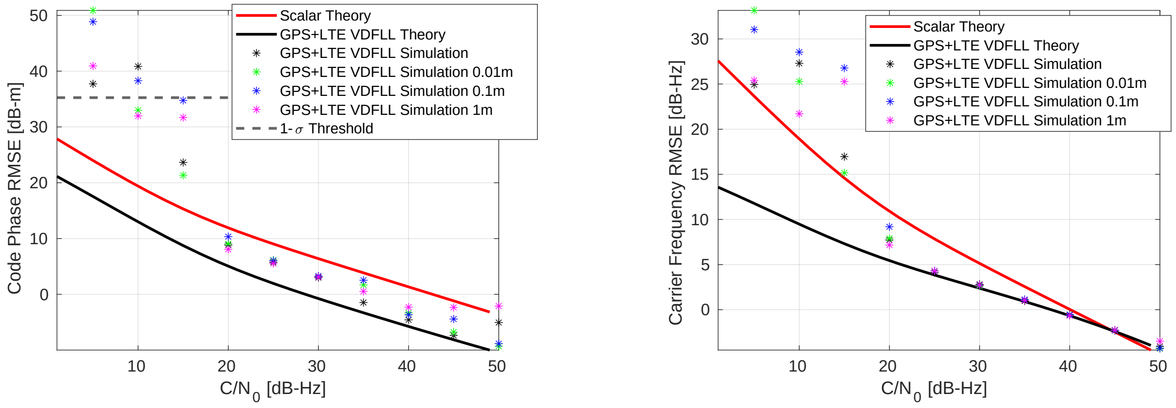

Figure A.3: 4 eNodeB dynamic simulation results. The LTE signals are tracked using the SSS.

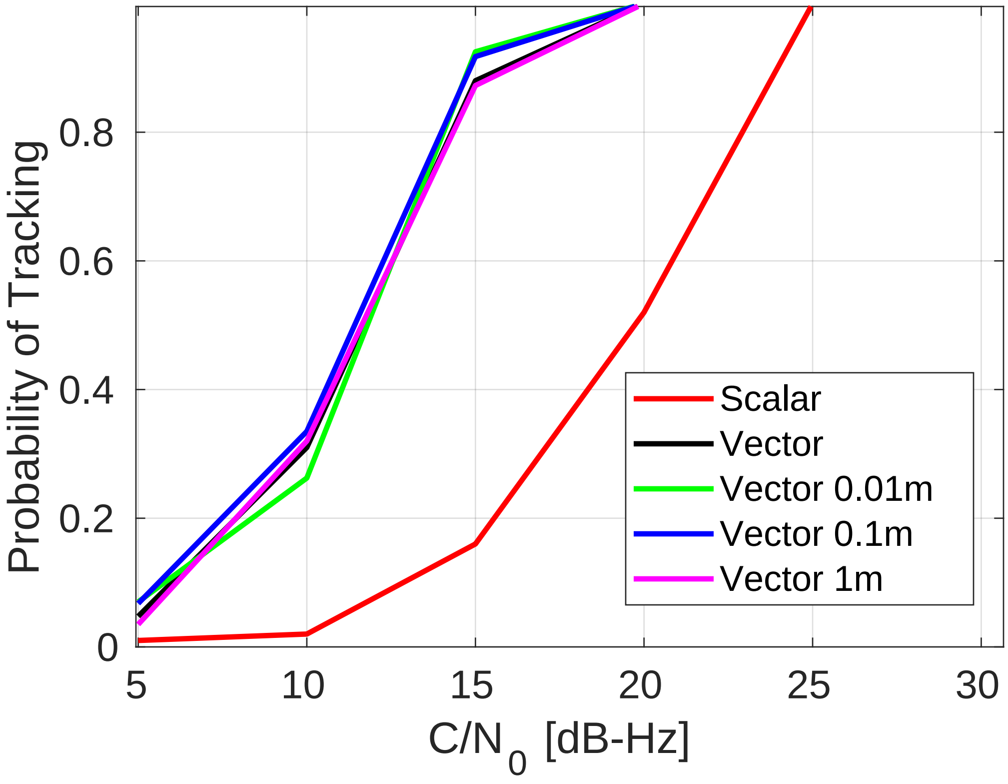

Figure A.4: Probability of tracking LTE in the 4 eNodeB static simulations. The LTE signals are tracked using the SSS.

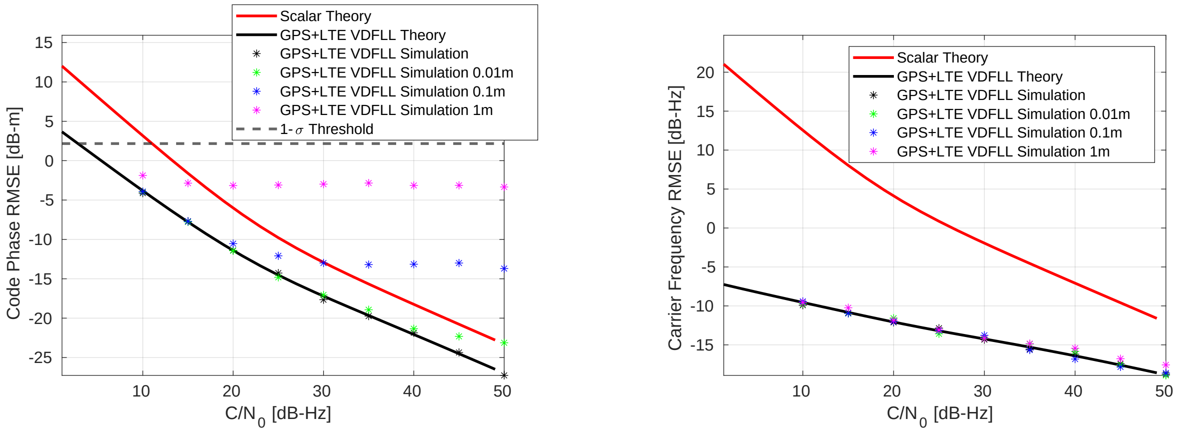

Figure A.5: 4 eNodeB static simulation results. The LTE signals are tracked using the CRS.

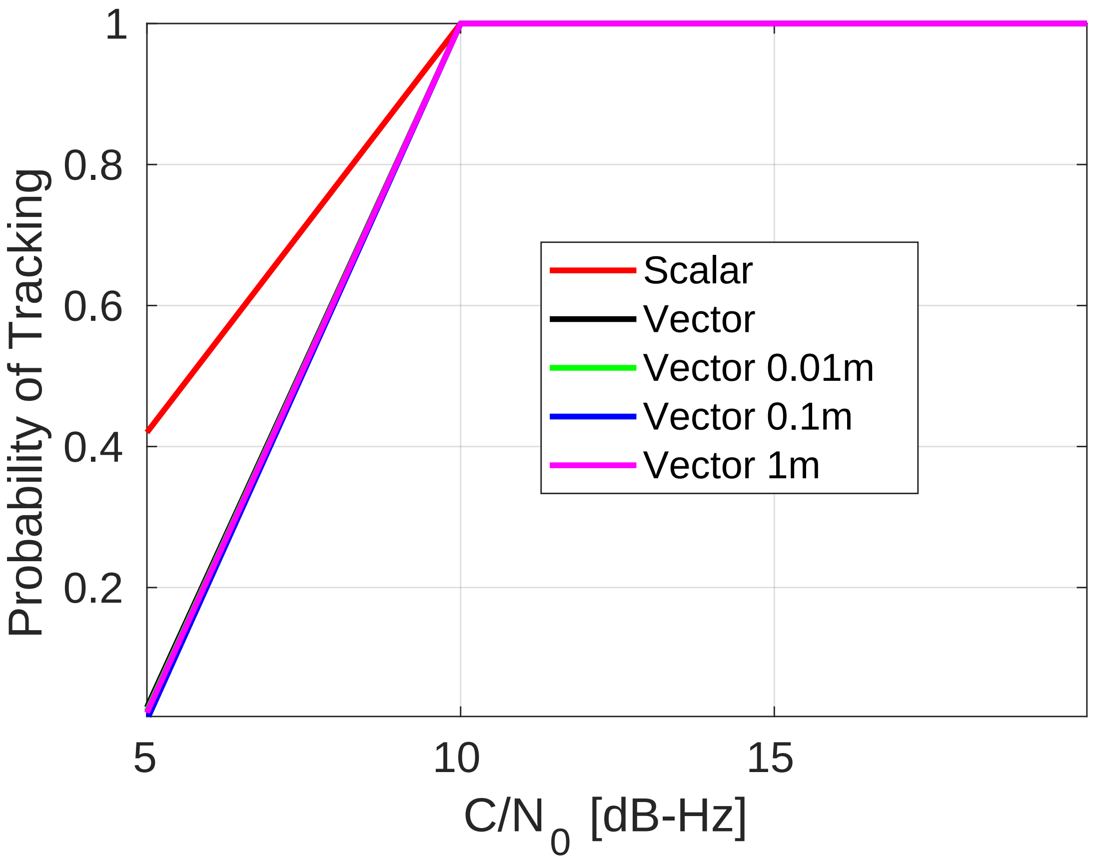

Figure A.6: Probability of tracking LTE in the 4 eNodeB static simulations. The LTE signals are tracked using the CRS.

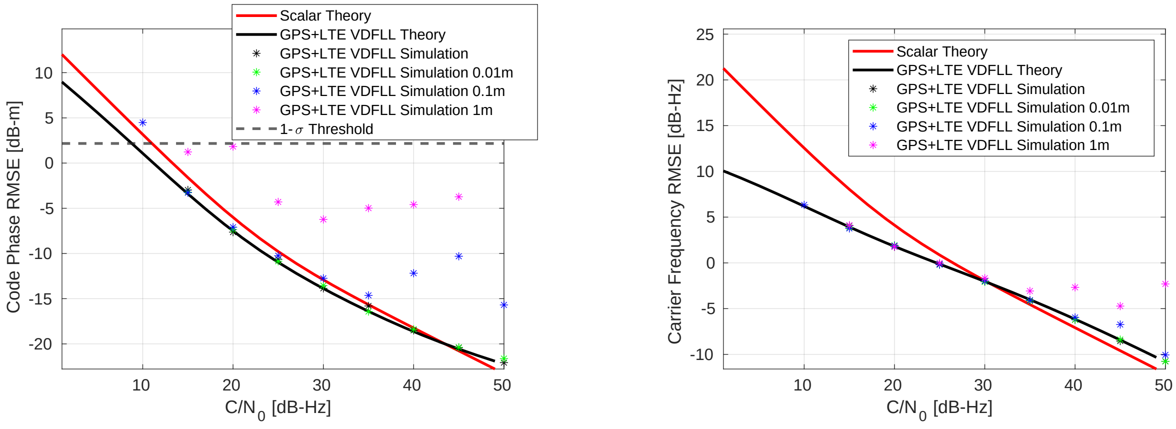

Figure A.7: 4 eNodeB dynamic simulation results. The LTE signals are tracked using the CRS.

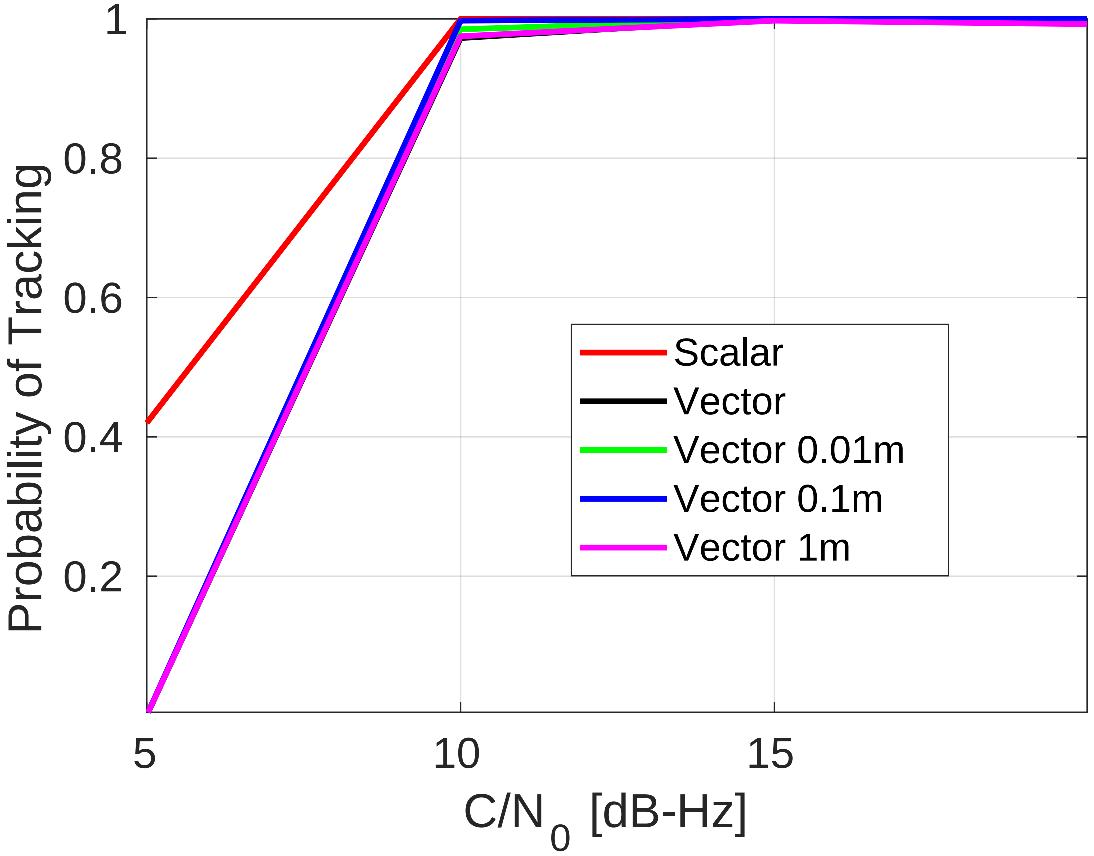

Figure A.8: Probability of tracking LTE in the 4 eNodeB static simulations. The LTE signals are tracked using the CRS.

Appendix B

4 eNodeB Multipath Simulation Results

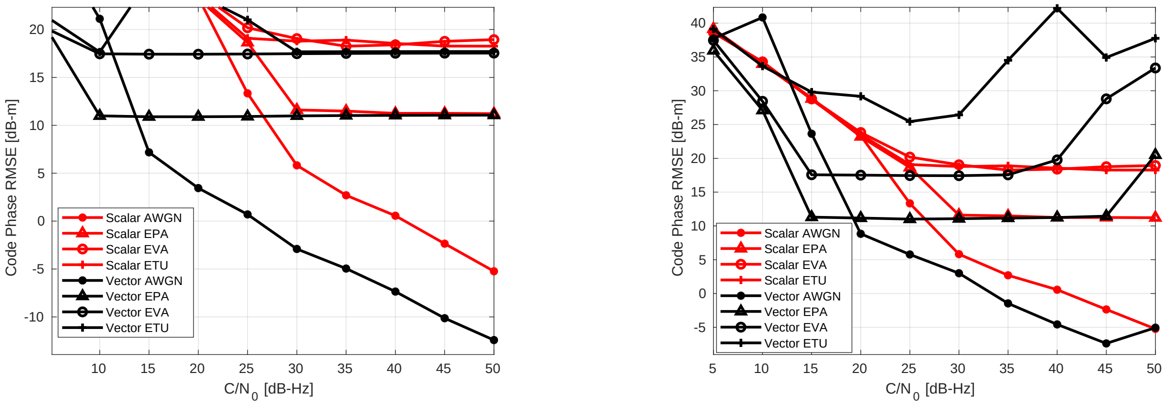

Figure B.1: The code phase tracking performance of the 4 eNodeB static simulations (left) and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS

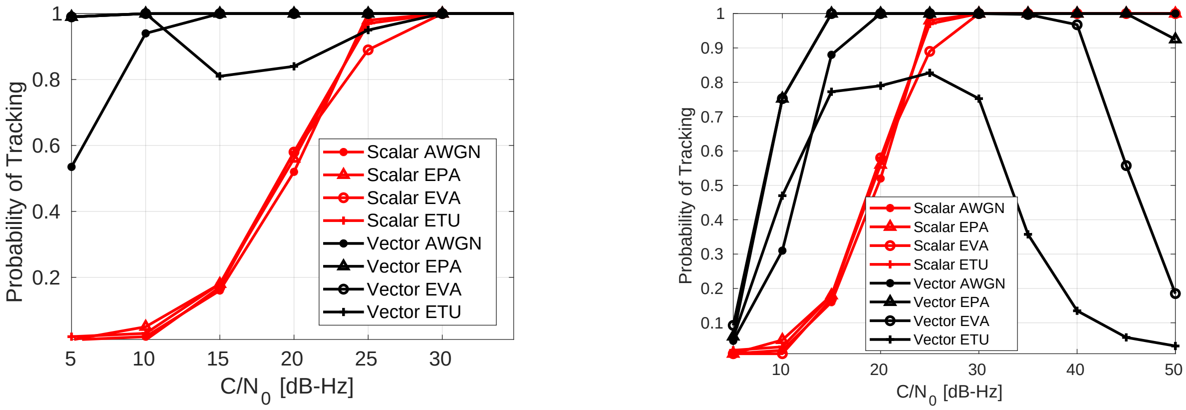

Figure B.2: The probability of tracking the LTE signal during the 4 eNodeB static simulations (left) and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the SSS.

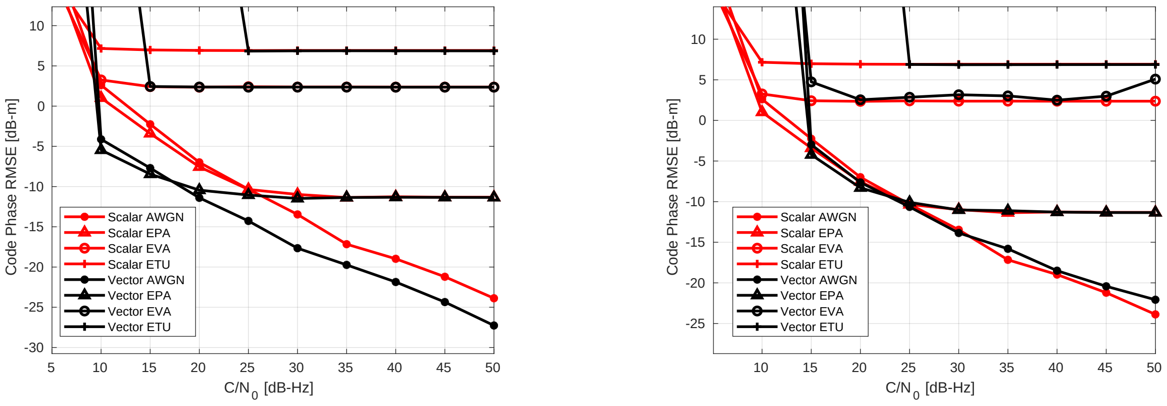

Figure B.3: The code phase tracking performance of the 4 eNodeB static simulations (left) and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS.

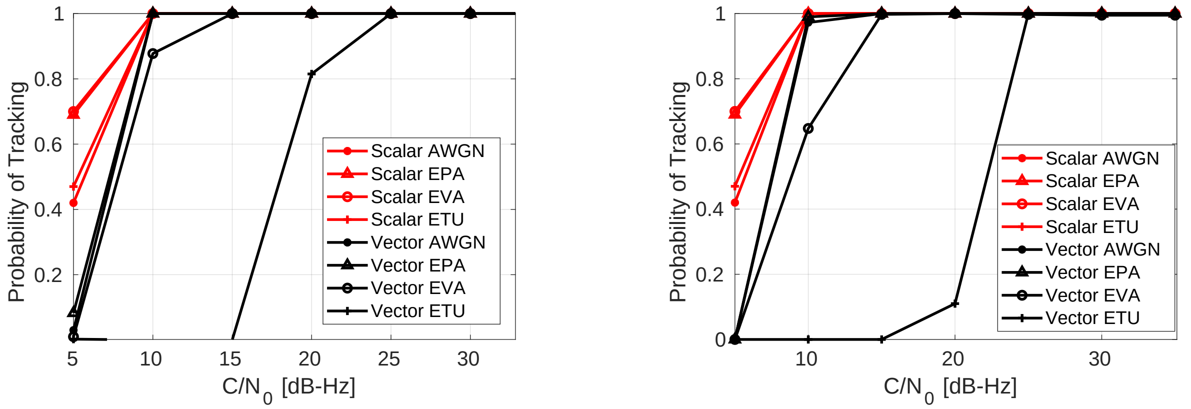

Figure B.4: The probability of tracking the LTE signal during the 4 eNodeB static simulations (left) and 4 eNodeB dynamic simulations (right) in the presence of multipath. The LTE signals are tracked using the CRS.